

普通高等教育“十一五”国家级规划教材

电子信息科学与工程类专业精品教材

多媒体通信技术基础

（第三版）

蔡安妮　等编著

電子工業出版社

Publishing House of Electronics Industry

北京 • BEIJING

内 容 简 介

本书是普通高等教育“十一五”国家级规划教材。作者紧密跟踪国际上多媒体技术发展的动向和研究成果，在分析本领域内大量具有代表性的文献及书籍的基础上，结合多年的科研和教学经验，综合提炼出本书的大纲和内容。全书比较全面地介绍了这一新领域内的主要理论与技术，包括：多媒体技术的特征、视觉特性与彩色电视信号、数据压缩的基本技术、视频数据的压缩编码、音频数据的压缩编码、多媒体同步、多媒体传输网络、多媒体通信终端与系统、视频在分组网上的传输、视频在异构环境下的传输等 10 章。全书在理论上力求严谨、叙述上尽量深入浅出。

本书选材兼顾到研究生及本科生教学两个方面的教学需要，同时可以作为从事通信、信息及相关行业的科研和工程技术人员的参考。

未经许可，不得以任何方式复制或抄袭本书之部分或全部内容。
版权所有，侵权必究。

图书在版编目(CIP)数据

多媒体通信技术基础 / 蔡安妮等编著. — 3 版. —北京：电子工业出版社，2012.11
电子信息科学与工程类专业精品教材
ISBN 978-7-121-18903-6

Ⅰ. ①多… Ⅱ. ①蔡… Ⅲ. ①多媒体通信－通信技术－高等学校－教材 Ⅳ. ①TN919.85

中国版本图书馆 CIP 数据核字(2012)第 267647 号

策划编辑：陈晓莉
责任编辑：陈晓莉
印　　刷：
装　　订：
出版发行：电子工业出版社
　　北京市海淀区万寿路 173 信箱　邮编 100036
开　　本：787×1092　1/16　印张：21.5　字数：780 千字
印　　次：2012 年 11 月第 1 次印刷
印　　数：3000 册　定价：46.00 元

凡所购买电子工业出版社图书有缺损问题，请向购买书店调换。若书店售缺，请与本社发行部联系，联系及邮购电话：(010)88254888。

质量投诉请发邮件至 zlts@phei.com.cn，盗版侵权举报请发邮件至 dbqq@phei.com.cn。

服务热线：(010)88258888。

第三版前言

时隔四年，本书第三版出版了。在这一版中，根据技术的发展，删去了一些过时的内容，补充了一些新概念和新技术，并增加了少量习题。此外在选材和叙述上，力求更为简明和精练，以突出重点内容。

与本书第二版相比，第三版的主要改动如下：

由于多媒体应用，特别是视频应用的迅速发展和普及，在第 1 章中删去了对多媒体早期业务的叙述。

由于模拟电视正在被数字电视逐步取代，在第 2 章中删去了复合电视信号的数字化一节。

第 3 章的改动主要在对小波变换讲述的精练和静止图像变换系数量化的删节。

第 4 章针对基于帧和基于对象的编码方法分别增加了码流结构一节，使内容更加完整，并在压缩编码的国际标准中补充了对正在制定的 HEVC 标准的简介。同时，将压缩算法的性能评价一节从第 3 章移至第 4 章，加入了对视频压缩率失真性能的定量评价方法。

在第 6 章中删去了多媒体同步的 4 层参考模型一节。

第 7 章的改动比较大，删去了有关物理层和链路层的技术，仅保留了重要网络的基本工作原理和性能（包括 QoS 机制）。同时，引入了用户体验质量（QoE）的概念，并将无线局域网、移动网和 WiMAX 合并到一节，因为这三种网络正在向融合的方向发展。

在第 8 章中删去了一些使用不多的传输层协议以及有关 ATM 网的视听业务标准，因为 ATM 技术将不会在终端上应用。此外，流媒体文件一节中改用使用更为广泛的 MP4 作为例子进行讲解；删去了流式应用系统的关键技术一节，因为相关内容已包含在其他章节中。更为重要的是，本章最后增加了两节：P2P 流式应用系统和 2011 年底 MPEG 制定标准的动态自适应 HTTP 流式应用系统（DASH），这是两种正在和将会成为主流的流媒体技术。

第 9 章删去了服务质量的保障和循环冗余码两节，因为它们主要涉及传输层以下的技术；同时，补充了无线网络中的拥塞控制一节。

第 10 章删除了视频信号模型一节。

最后，向给本书第二版提出意见和建议的老师和学生表示衷心的感谢！

作　者

2012 年 7 月

再版前言

本书在2000年出版的“多媒体通信技术基础”第一版的基础上，根据近年来技术的发展和曾经使用过该书的教师和学生的意见，经过较大幅度的增删和修改而成。本书与第一版一样，作者是按照下述方向努力的：

首先阅读这个新技术领域中绝大多数有代表性的论文、书籍，并根据自己的科研经验，经过消化、提炼，分门别类，使其系统化；然后用自己的语言，尽量通俗、严谨、简繁适当地写出来。

本书选材时尽量兼顾了本科生教学和研究生教学两方面的需要，以及在通信、信息及相关领域中从事研究开发的工程技术人员掌握多媒体技术的需要。在高等院校使用本教材时，可根据本校、本系的特点，选择书中的不同章节、或不同章节中的部分内容分别作为本科生、研究生讲课使用。为了加深理解和拓展各章中的内容，每章之后都附有一定数量的习题和参考文献。

本书第1、2章保留由孙景鳌在第一版中编写的内容；第5章由苏菲编写；3.8节、4.2、4.3.3节和10.2、10.3节由庄伯金编写。

本书取材于众多的文献，作者在此对这些推动多媒体技术发展的人们表示敬意和感谢；同时，对给本书第一版提出意见和建议的老师和学生表示感谢。最后，作者仅以本书缅怀北京邮电大学多媒体通信与模式识别实验室创建人、本书第一版作者之一孙景鳌教授，由于他在实验室对多媒体方向科研的倡导和坚持，才有今天这本书。

作　者

2008年3月

目　录

第 1 章　概论——多媒体技术的特征 …… 1
1.1　概述 …… 1
1.2　多媒体的概念与含义 …… 1
1.3　多媒体产生的技术背景 …… 3
1.3.1　图像压缩编码技术的成熟 …… 3
1.3.2　大规模集成电路技术的发展 …… 4
1.3.3　大容量数字存储技术的发展 …… 4
1.4　多媒体系统的基本类型及相关业务 …… 5
1.4.1　独立商亭式系统 …… 5
1.4.2　多媒体信息检索与查询 …… 5
1.4.3　多媒体会议与协同工作 …… 7
1.4.4　多媒体即时通信 …… 9
1.4.5　点播电视(VOD) …… 10
1.5　三网融合及相应的业务 …… 11
1.5.1　网络的融合 …… 11
1.5.2　多重服务与业务融合 …… 13
习题一 …… 13
参考文献 …… 13
第 2 章　视觉特性和彩色电视信号 …… 14
2.1　人的视觉特性 …… 14
2.1.1　图像对比度与视觉的对比度灵敏度特性 …… 14
2.1.2　空间频率与视觉的空间频率响应 …… 15
2.1.3　视觉的时间域响应 …… 16
2.1.4　彩色的计量和彩色视觉 …… 17
2.2　彩色电视信号 …… 19
2.2.1　扫描——空间频率到时间频率的转换 …… 19
2.2.2　隔行扫描与逐行扫描 …… 19
2.2.3　电视信号的带宽 …… 20
2.2.4　彩色空间的处理 …… 21
2.2.5　全彩色电视信号 …… 22
2.3　分量彩色电视信号的数字化 …… 23
习题二 …… 25
参考文献 …… 26
第 3 章　数据压缩的基本技术 …… 27
3.1　概述 …… 27
3.2　数据压缩的理论依据 …… 27
3.2.1　离散信源的信息熵 …… 27
3.2.2　信源的概率分布与熵的关系 …… 28
3.2.3　信源的相关性与序列熵的关系 …… 29
3.3　信息率—失真理论 …… 30
3.3.1　通信系统的一般模型 …… 30
3.3.2　信息率—失真函数 …… 31
3.3.3　限失真信源编码定理 …… 32
3.4　取样频率的转换 …… 33
3.4.1　下取样 …… 33
3.4.2　上取样 …… 35
3.4.3　分数比率转换 …… 37
3.5　预测编码 …… 37
3.5.1　差分脉冲编码调制(DPCM) …… 37
3.5.2　序列图像中运动矢量的估值 …… 39
3.5.3　具有运动补偿的帧间预测 …… 44
3.6　正交变换编码 …… 46
3.6.1　最佳线性正交变换 …… 46
3.6.2　离散余弦变换 …… 47
3.7　子带编码 …… 50
3.7.1　子带编码的工作原理 …… 50
3.7.2　正交镜像滤波器组 …… 52
3.7.3　时域混叠消除 …… 53
3.8　小波变换编码 …… 56
3.8.1　多尺度分析 …… 56
3.8.2　二进小波变换 …… 57
3.8.3　变换系数的排序和编码 …… 58
3.9　量化 …… 61
3.9.1　均匀量化器 …… 61
3.9.2　最小均方误差量化器 …… 62
3.9.3　最小熵量化器 …… 63
3.9.4　自适应量化 …… 64
3.9.5　预测误差和 DCT 系数的量化 …… 64
3.10　熵编码 …… 65

3.10.1 熵编码的基本概念 …… 65
3.10.2 霍夫曼编码 …… 66
3.10.3 算术编码 …… 68
习题三 …… 71
参考文献 …… 74
第 4 章 视频数据的压缩编码 …… 75
4.1 基于帧的视频编码 …… 75
4.1.1 典型的编码器与解码器 …… 75
4.1.2 视频序列的编码 …… 79
4.1.3 帧内预测编码 …… 80
4.1.4 帧间预测编码的优化 …… 81
4.1.5 基于率－失真优化的编码模式选择 … 86
4.1.6 低计算复杂度的变换编码与量化 …… 88
4.1.7 码流结构 …… 91
4.2 基于对象的视频编码 …… 92
4.2.1 编码器与解码器 …… 92
4.2.2 任意形状 VOP 的预测编码 …… 93
4.2.3 任意形状 VOP 的纹理编码 …… 94
4.2.4 形状编码 …… 95
4.2.5 码流结构 …… 96
4.3 恒定速率编码器的速率控制 …… 96
4.3.1 高码率应用的速率控制(TM 5) …… 96
4.3.2 速率控制中的率－失真模型 …… 98
4.3.3 低延时应用的速率控制(TMN8) … 100
4.3.4 基于率—失真优化的速率控制(JVT—G012) …… 102
4.3.5 视频缓存证实器 …… 105
4.4 变速率视频编码 …… 106
4.5 压缩编码算法性能的评价 …… 107
4.6 图像和视频压缩编码的国际标准 …… 109
4.6.1 静止图像压缩编码标准 JPEG/JPEG2000 …… 109
4.6.2 视听会议压缩编码标准 H.261 …… 110
4.6.3 数字声像存储压缩编码标准 MPEG－1 …… 110
4.6.4 通用视频及伴音压缩编码标准 MPEG－2(H.262) …… 111
4.6.5 低比特率视听会议压缩编码标准 H.263 …… 111
4.6.6 通用音视频对象压缩编码标准 MPEG－4 …… 112
4.6.7 视频压缩编码标准 H.264/AVC …… 113
4.6.8 视频压缩编码的国家标准——AVS …… 114
4.6.9 正在制定的视频压缩编码标准——HEVC …… 114
习题四 …… 115
参考文献 …… 117
第 5 章 音频数据的压缩编码 …… 118
5.1 概述 …… 118
5.2 人的听觉特性 …… 118
5.2.1 响度级和响度 …… 118
5.2.2 听觉灵敏度 …… 119
5.2.3 听觉掩蔽 …… 120
5.2.4 临界带宽 …… 121
5.3 音频信号的数字化 …… 121
5.4 音频自适应差分脉冲编码调制 …… 122
5.5 音频子带编码 …… 122
5.6 线性预测编码 …… 124
5.7 码激励线性预测编码(CELP) …… 124
5.7.1 感知加权滤波器 …… 125
5.7.2 合成分析法 …… 125
5.7.3 CELP 编解码原理 …… 126
5.7.4 G.729 编解码器 …… 127
5.8 感知编码 …… 128
5.9 MPEG-1 音频编码 …… 129
5.9.1 概述 …… 129
5.9.2 MPEG-1 心理声学模型 …… 130
5.9.3 编码层次 …… 131
5.10 MPEG-2 音频编码 …… 134
5.10.1 MPEG-2 BC …… 134
5.10.2 MPEG-2 AAC …… 135
5.11 杜比(AC-3)编码 …… 137
5.12 音频压缩编码的国际标准 …… 139
习题五 …… 141
参考文献 …… 141
第 6 章 多媒体同步 …… 142
6.1 多媒体数据 …… 142
6.1.1 连续媒体数据与静态媒体数据 …… 142
6.1.2 多媒体数据内部的约束关系 …… 143
6.1.3 多媒体数据的构成 …… 144
6.2 多媒体数据时域特征的表示 …… 145
6.2.1 时域场景和时域定义方案 …… 145

6.2.2 时域参考框架 …… 146
6.2.3 描述时域特征的时间模型 …… 147
6.2.4 同步容限 …… 151
6.3 分布式多媒体系统中的同步 …… 151
6.3.1 分布式多媒体系统的结构 …… 151
6.3.2 同步规范的传送 …… 152
6.3.3 影响多媒体同步的因素 …… 152
6.3.4 多级同步机制 …… 154
6.4 连续媒体内部的同步 …… 154
6.4.1 基于播放时限的同步方法 …… 154
6.4.2 基于缓存数据量的同步方法 …… 156
6.5 媒体流之间的同步 …… 157
6.5.1 基于全局时钟的时间戳方法 …… 158
6.5.2 基于反馈的流间同步方法 …… 159
6.6 接收与发送时钟的同步 …… 160
6.6.1 基于接收缓存器的方法 …… 160
6.6.2 基于时间戳的锁相方法 …… 161
6.6.3 基于网络时间协议的方法 …… 162
6.7 同步算法小结 …… 162
习题六 …… 163
参考文献 …… 164
第 7 章 多媒体传输网络 …… 165
7.1 概述 …… 165
7.2 多媒体信息传输对网络的要求 …… 165
7.2.1 性能指标 …… 165
7.2.2 网络功能 …… 168
7.2.3 服务质量(QoS)和用户体验质量(QoE)的保障 …… 169
7.3 网络类别 …… 170
7.3.1 电路交换网络和分组交换网络 …… 171
7.3.2 面向连接方式和无连接方式 …… 172
7.3.3 资源预留、资源分配和资源独享 … 172
7.4 电路交换广域网对多媒体信息传输的支持 …… 173
7.4.1 电路交换广域网 …… 173
7.4.2 多点控制单元 …… 174
7.5 分组交换广域网对多媒体信息传输的支持 …… 175
7.5.1 帧中继 …… 175
7.5.2 异步传输模式 ATM …… 175
7.6 局域网对多媒体信息传输的支持 …… 177
7.6.1 共享介质的局域网 …… 177
7.6.2 吉比特以太网 …… 179
7.6.3 以太网 QoS 保障 …… 180
7.7 IP 网对多媒体信息传输的支持 …… 181
7.7.1 传统的 IP 网(IP v4) …… 181
7.7.2 IP 多播 …… 183
7.7.3 新一代 IP(IP v6) …… 184
7.8 IP QoS 保障机制 …… 186
7.8.1 综合服务模型与资源预留协议 RSVP …… 186
7.8.2 区别服务 …… 187
7.8.3 多协议标记交换 MPLS …… 189
7.8.4 流量工程和基于约束的寻径 …… 191
7.9 无线网络对多媒体信息传输的支持 …… 192
7.9.1 无线传输的特点 …… 192
7.9.2 无线局域网 …… 193
7.9.3 蜂窝移动通信网 …… 198
7.9.4 无线城域网(WiMAX) …… 202
7.9.5 无线网络中多媒体传输的特殊问题 … 204
7.10 宽带用户接入网 …… 205
7.10.1 数字用户线路 …… 205
7.10.2 光缆接入 …… 207
7.10.3 光缆－同轴电缆混合接入 …… 208
7.11 下一代网络 …… 213
7.11.1 网络融合模型 …… 213
7.11.2 IP 多媒体子系统(IMS) …… 213
7.11.3 下一代网络框架 …… 214
习题七 …… 215
参考文献 …… 216
第 8 章 多媒体通信终端与系统 …… 217
8.1 传输层协议 …… 217
8.1.1 应用层分帧和集成层次处理 …… 217
8.1.2 传统的因特网传输层协议 …… 218
8.1.3 实时传输层协议 RTP 及 RTCP … 219
8.1.4 流控制传输协议(SCTP) …… 222
8.2 多媒体数据的复接与分接 …… 223
8.2.1 同步时分复用 …… 223
8.2.2 分组复用 …… 224
8.2.3 多媒体数据的封装 …… 229
8.3 视听通信系统与终端 …… 231
8.3.1 视听通信终端的一般框架 …… 231
8.3.2 同步电路交换网视听业务标准(H.320) …… 232

8.3.3 分组交换网视听业务标准(H.323) …… 233
8.3.4 公用电话网视听业务标准(H.324) …… 235
8.3.5 移动电话网视听业务标准(H.324M 和 3G-324M) …… 236
8.3.6 视听系统的通信控制协议 …… 238
8.3.7 H 系列系统间的互通 …… 241
8.4 会话发起协议和会话描述协议 …… 242
8.4.1 会话发起协议(SIP) …… 242
8.4.2 SIP 消息格式和会话描述协议(SDP) …… 245
8.4.3 SIP 与 H.323 信令的比较 …… 246
8.5 协同计算与组通信 …… 246
8.5.1 协同计算的概念 …… 246
8.5.2 组通信 …… 247
8.5.3 应用共享控制 …… 248
8.6 多媒体流式应用系统与终端 …… 250
8.6.1 流媒体的概念 …… 250
8.6.2 多媒体流式应用系统的结构 …… 250
8.6.3 实时流协议(RTSP) …… 253
8.6.4 流媒体文件 …… 254
8.6.5 流服务器 …… 256
8.6.6 内容分发网络 …… 260
8.6.7 3GPP 分组交换流服务(PSS) …… 264
8.6.8 IPTV …… 265
8.7 对等网络(P-to-P)流式应用系统 …… 270
8.7.1 工作原理 …… 270
8.7.2 网络结构与数据传输 …… 271
8.7.3 其他设计原则 …… 272
8.8 动态自适应 HTTP 流式应用系统 …… 274
8.8.1 HTTP 流传输技术的发展 …… 274
8.8.2 系统的工作原理 …… 274
8.8.3 MPEG DASH 标准 …… 276
习题八 …… 278
参考文献 …… 278
第 9 章 视频数据的分组传输 …… 280
9.1 概述 …… 280
9.2 应用层拥塞控制 …… 280
9.2.1 TCP 拥塞控制 …… 280
9.2.2 TCP 流量模型和 TCP 友好性 …… 281
9.2.3 基于速率的拥塞控制 …… 282
9.2.4 拥塞检测 …… 283
9.2.5 无线网络中的拥塞控制 …… 285
9.3 传输层差错控制 …… 286
9.3.1 自动重传技术(ARQ) …… 286
9.3.2 前向纠错码 …… 287
9.3.3 前向删除复原码 …… 291
9.3.4 数据交错 …… 296
9.3.5 前向差错控制与 ARQ 的结合 …… 296
9.4 编码层差错控制 …… 297
9.4.1 误码和包丢失对已压缩视频信号的影响 …… 297
9.4.2 抗误码编码 …… 298
9.4.3 错误掩盖 …… 305
9.4.4 编解码器交互差错控制 …… 309
9.5 差错控制小结 …… 311
习题九 …… 312
参考文献 …… 314
第 10 章 视频在异构环境中的传输 …… 315
10.1 概述 …… 315
10.2 可伸缩性编码 …… 316
10.2.1 可伸缩性编码的概念 …… 316
10.2.2 空间域可伸缩性编码 …… 317
10.2.3 时间域可伸缩性编码 …… 318
10.2.4 质量可伸缩性编码 …… 320
10.2.5 可伸缩性编码的联合应用 …… 321
10.2.6 基于小波变换的可伸缩性编码 … 322
10.3 视频转码 …… 322
10.3.1 视频转码框架 …… 323
10.3.2 运动向量的再利用 …… 325
10.3.3 DCT 域的运动补偿 …… 327
10.3.4 转码的速率控制 …… 328
10.4 多描述视频编码 …… 330
10.4.1 多描述编码的概念 …… 330
10.4.2 多描述的生成 …… 331
10.4.3 视频多描述编码 …… 335
习题十 …… 336
参考文献 …… 336

第1章　概论——多媒体技术的特征

1.1　概述

在技术发展史上，计算机、通信和广播电视一直是三个互相独立的技术领域，各自有着互不相同的技术特征和服务范围。但是，近几十年来，随着数字技术的发展，这三个原本各自独立的领域相互渗透、相互融合，形成了一门崭新的技术——多媒体。多媒体技术的最初体现是配之以声卡、视卡的多媒体计算机。它一出现立即在世界范围内，在家庭教育和娱乐方面得以广泛的应用，并且迅速地向网络化的远程应用发展。多媒体技术的应用与发展，又反过来进一步加速了这三个领域的融合，使多媒体通信成为通信技术今后发展的主要方向之一。

有许多技术，从它们开始出现时就给人以清楚明了的概念。例如电话技术，从最初用两根电线把两部简陋的电话机连接起来实现远距离通话时起，它就被称为电话技术；后来经历了人工交换、步进制交换、程控交换，以至于发展到数字式移动电话，仍然是电话技术。多媒体技术所遇到的情况则有所不同。使人们不容易清楚地建立起“什么是多媒体”的概念的因素很多。首先，通信、计算机与彩色电视本来都是技术面宽而复杂的技术，由它们融合在一起而产生的多媒体技术，其技术覆盖面自然就更宽，技术的交叉更为复杂。这就使得多媒体不能像其他诸如电话、电影、电视、汽车、马车等事物那样一目了然。另外，为了经济上或商业上的利益，某些商家把本来不属于多媒体的技术说成是多媒体技术，人为地造成了概念上的混乱。此外，新闻报道中某些不准确的用词也产生了概念上的误导。由于上述种种原因，造成了这样的局面：“如果你向10个不同的人请教多媒体一词的含义，你至少会得到10种不同的答案”[1]。

鉴于上述情况，我们力图在本章中，使读者对多媒体和多媒体技术建立起一个比较完整和全面的概念。

1.2　多媒体的概念与含义

1984年美国RCA公司在普林斯顿的David Sarnoff实验室，组织了包括计算机、广播电视和信号处理三个方面的40余名专家，综合了前人已经取得的科研成果，经过4年的研究，将彩色电视技术与计算机技术融合在一起，于1987年在国际第二届CD-ROM年会上展出了世界上第一台多媒体计算机。这项技术后来定名为DVI(Digital Video Interactive)。这便是**多媒体**(Multimedia，由Multiple Media两个词组合而成)技术的雏形。

多媒体技术一出现，在世界范围内立即引起巨大的反响，因为它清楚地展现出信息处理与传输(即通信)技术的革命性的发展方向。国际上在同一年内立即成立了交互声像工业协会，当该组织1991年更名为交互多媒体协会IMA(Interactive Multimedia Association)时，已经有15个国家的200多个公司加入了。

多媒体计算机与普通计算机有什么不同，它的出现何以如此引人瞩目？早期的计算机只能进行数学运算，后来又具有了文字处理能力，经过若干年的发展之后又增加了图形与动画的功能。而多媒体计算机则是增加了对包括伴音在内的活动图像(即动作连续的电视图像)的处理、存储和显示的能力。这在技术上是一个质变性的飞跃。

为了比较深入地理解这一飞跃，首先让我们来作一个简单的计算。我国的电视制式在一帧图像内有625行，去掉在扫描逆程（不显示图像的部分）中占去的约50行，出现在一幅画面上有效的扫描行数是576行左右。电视画面的宽与高尺寸之比是4∶3，要保证图像在水平方向上单位距离内可分辨的像素数与垂直方向上相等，那么在图像水平方向上的像素数应为576×4/3＝768。将这样一幅单色的电视图像数字化，其取样点数应不少于768×576。

根据三基色原理，一幅彩色图像是由红、绿、蓝3幅单色图像组成的，每秒钟要传送25幅彩色图像才能保证电视图像的连续性。如果每一个像素采用8比特量化，1秒钟时间内需要传送的数据量则为

$$768 \times 576 \times 3 \times 8 \times 25 \approx 265\ (\text{Mb}) = 33\ (\text{MB})$$

1部2小时电视电影的总数据量为

$$33 \times 60 \times 60 \times 2 = 238\ (\text{GB})$$

在20世纪80年代，计算机硬盘的容量是40 MB，总线传输能力在3Mb/s左右，由此可见，普通微机不具有存储未经压缩的活动图像信号的能力，也无法通过总线将其送到显示器上进行连续显示。

多媒体计算机的出现，标志着人们已经在电视图像信号（或称视频信号）的实时压缩（使得能在实时的条件下存储、传送活动图像）、实时解压缩（以显示活动图像）等技术上取得了突破性的进展，具有划时代的意义。

多媒体计算机区别于普通计算机的第二个技术特征，是解决了同时存储、读取和显示两个在时间上紧密相关的数字信号（即伴音信号和图像信号）时，如何在时间上保持**同步**的问题。同步的必要性很容易理解，屏幕上说话的人嘴形与声音配不上就是不同步。在电视系统中，图像信号与伴音信号是组合成一个信号传送的，二者之间总是保持着同步。在多媒体计算机中，数字形式的伴音与视频图像信号是可以作为两个信号分别存储的，要保证同步就需要考虑二者在读取、处理和显示过程中的正确时间关系。在单纯处理数字、文字和图形的普通计算机中并不需要维护这种严格的时间约束关系。当多媒体扩展到远程应用时，服务器与客户端（或两个对等的终端）通过网络连接，由于网络延时和延时抖动的存在，保证声/像信号在传输过程中的同步也是有别于传统数据传输的新问题。

根据以上分析我们看到，多媒体计算机所增加的处理包括伴音在内的电视图像的功能，不仅使计算机在功能数量上多了一项，而是一个质变的过程。声音和电视图像在传统上是属于通信技术和电视技术的研究对象的，多媒体计算机已经将计算机延伸到电视技术和通信技术领域了。

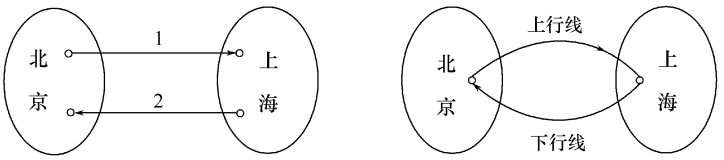
前面的简单计算表明，包括伴音在内的电视信号，是各种媒体中数据量最大的一种。既然技术已经发展到了能用计算机处理这种数据量最大的业务，其他数据量小的，如信息查询、电子邮件、电话、传真、可视电话等项业务，都将可以融合到一起，在一台多媒体终端上实现。这是多媒体中“多”字的另一层含意。也有人用**集成**一词来描述多媒体将多种媒体、多项业务集合在一起的这一特点。

有的学者从数据本身的特点来理解多媒体技术与其他技术的区别。多媒体数据是由内容上相互关联的文本、图形、图像、声音、动画、活动图像等媒体的数据所形成的**复合数据**。这一数据合成的过程是在计算机控制下完成的[2]。在这里首先说明了多媒体所涉及的信号是数字化的，而不是模拟的；其次应当注意“内容上相互关联”和“合成”这两个要点，内容上毫不相干的文字、图形、声音、活动图像等数据的集合并不是多媒体数据。通常对时间敏感的声音、活动图像的数据称为**实时数据**，其他类型的数据则是**非实时数据**。如果说某个数据是多媒体数据，则意味着该复合数据中至少包含有一种实时数据和一种非实时数据[3]。

能够通过计算机对视频数据进行操作，形成了多媒体技术的另一个基本特征，即**交互性**（Interactive）。人—机交互是计算机固有的技术特征。通过键盘或鼠标打开一个文件，就会显示你所要的内容，经过修改、补充之后，再写入到给定的文件名下，或者通过一条打印指令则可在打印机上将

文章打印出来。这种人与计算机之间通过指令“对话”式的操作就是交互操作，这已经是人们几十年来所熟知的常识。而多媒体强调的是以交互的方式对活动图像（包括伴音）进行操作，例如，对其进行“快进”、“暂停”等录像机式的操作。

值得注意，人—机交互与传统的通信领域中的**双向通信**（Two Ways）是不同的两个概念。双向通信意味着通信可以在两个方向上进行，如图 1-1（a）所示，北京的电视节目经过线路 1 传到上海，上海的电视节目经过线路 2 传到北京，二者使用了对称的信道，但相互独立。所谓对称是指两条线路的信道带宽或数据率等传输参数与指标是相同的。而图（b）表示由北京的通信终端自动地从位于上海的数据库内调取电视节目时的交互工作。这项业务使用了双向的非对称信道，**上行线**（由用户向信息中心）传送指令，是低速率的；**下行线**传送读取的电视信号，是高速率的。需要强调的是，这项业务中的上、下行信道中传输的信息是相互关联的，体现了人—机交互操作。



（a）双向通信　　（b）交互通信

图 1-1　交互工作与双向通信

除了人—机交互之外，一些多媒体系统支持人与人之间的交互，例如，可视电话、多媒体会议和协同工作就是典型的例子。传统的电话也是实现人与人交互的通信系统；支持人与人之间交互的多媒体系统与电话的最大区别在于，它使用了更多的媒体（除相互听得见外还相互看得见）和允许同时进行更多的业务（如除通话外还可以共同观看、修改文件等）。

通过上面的讨论，我们可以对多媒体的特征作一个简要的总结，即它具有**集成性**、**同步性**和**交互性**。其中集成性包括多种媒体的集成和多种业务的集成；交互性包括人与人的交互和人—机交互。

1.3　多媒体产生的技术背景

一种新技术的产生与发展往往是与其特定的技术背景相联系的，是以其他有关技术的发展作为基础的。实际上，多媒体技术之所以能够在 20 世纪 80 年代末期出现，主要得益于下述几个方面的技术成果。

1.3.1　图像压缩编码技术的成熟

在通信领域中，人人都知道数字通信具有模拟信号通信所无法比拟的优越性。模拟信号在传输过程中产生了失真或者混进去噪声，在接收端难以使其恢复原形。数字信号则不同，因为发送的脉冲信号形状是已知的，如果在传输中产生失真或叠加上噪声，在接收端经过放大、幅度切割等整形处理，失真和噪声则被消除，又恢复成原来的形状。

数字通信的缺点是将模拟信号变为数字信号以后，对信道带宽的要求大幅度增加。以电话为例，一个模拟话路只需要 3.4 kHz 的带宽。变成数字信号时，取样频率取 8 kHz（根据取样定理，取样频率的数值最低不得低于被取样信号的最高频率的 2 倍），每个取样点采用 8 比特量化，一路数字电话的数据率则为 64 kb/s。当用二进制码传输时，每赫兹带宽最高只能传送 2 b/s（采用多进制码传输时，这个数字可以高一些）。可见一路电话从模拟传送改为数字传送，对信道带宽的要求提高了很多。彩色电视所遇到的情况则更为困难。我们在 1.2 节里已粗略地估算了彩色电视信号的数据率。按照国际标准，一路按分量进行编码的彩色电视信号（不包括伴音）的数据率 $R=216$ Mb/s，而

一路模拟彩色电视信号的带宽只有 6 MHz。正是由于这个原因，虽然早在 1937 年 A. Reeues 就发明了 PCM(脉冲编码调制)，但数字通信得到广泛的应用还是在 20 世纪 70 年代之后。

要以数字方式传输电视信号，必须解决数据率的压缩问题。人们对信源压缩编码已进行了几十年的深入研究，进入 20 世纪 80 年代，这项技术已经较为成熟，能够将数字电视信号数据率实时地压缩到 34 Mb/s 左右。这里所讲的技术的成熟是指压缩方法，而实时是指压缩与解压缩的速度跟得上 25 帧/秒、或 30 帧/秒的图像显示要求。处理速度的高低取决于用以实现压缩和解压缩的电子电路的集成化水平。

人们研究数字电视信号的压缩编码问题的最初出发点，是要解决电视信号长距离传输中的抗干扰(失真、噪声)问题，也就是说，将电视台要发出的信号数字化，然后压缩编码以求用较低的数据率传送，传到目的地以后(如从北京传到上海)，数字信号经过切割整形，再还原为模拟信号，送到本地发射机发射出去。因此在多媒体出现之前的几十年中，电视信号的压缩编码一直是针对通信领域中的应用的。

将图像压缩编码的研究成果应用到计算机领域则导致了新技术的产生。当 DVI 技术与世人见面时，它已经能够将图像信号和伴音信号压缩 100 倍以上(包括适当地将电视信号的图像分辨率降低)，其速率为 1.2～1.4 Mb/s，这使得活动图像数据能够在当时的计算机总线上传输，从而成为计算机可以处理的数据类型之一。同时也使得 1 张 CD -ROM 能记录 74 分钟的电视节目，如果数字电视信号没有经过压缩，则只能记录 30 秒钟的电视节目。

1.3.2 大规模集成电路技术的发展

代表大规模集成电路技术水平的主要参数之一是制作在芯片上的线的宽度。线宽做得越窄，一块芯片上能容纳的元件便越多，集成度也便越高。至 20 世纪 80 年代末，已经能在芯片上制作线宽小于 0.5 μm 的线了。在多媒体技术发展初期，CPU(如 80286)的处理能力还比较低，那时数据的压缩和解压缩运算要靠专用的芯片来完成。在 Intel 公司的 DVI 技术中，图像的压缩、解压缩是用 2 个芯片来完成的，其中每个包含有 26 万多个晶体管[4]。这个数字清楚地表明，要实时地将彩色电视信号的数据率压缩到几个 Mb/s 以下，电路的集成度不高是无法实现的。

让我们看一下这几个数字，286 CPU 只集成了 13.4 万只晶体管，386 CPU 则有 27.5 万只，发展到 486、586(P5)和 686(P6)时，CPU 内集成的晶体管数分别为 120 万、300 万和 350 万只。这些数字充分说明了大规模集成电路技术发展之迅速，从而也为多媒体技术的发展提供了良好的条件。要使多媒体终端的成本降低到普通家庭的购买力能接受的水平，使之能像现在电话机这样普及，在很大程度上也要有赖于大规模集成电路技术的发展。

1.3.3 大容量数字存储技术的发展

激光视盘(LVD，后称 LD)是 20 世纪 70 年代研究成功的，能够在 1 张直径 12 英寸的大盘上记录大约 30 分钟的电视节目。LVD 的出现最初并没有引起太大的重视，人们的注意力还集中在探讨究竟光盘机与磁带录像机哪一种技术更有发展前途。光盘记录技术于 1982 年被用来记录音乐、流行歌曲，1 张 5 英吋直径的小盘 CD(Compact Disc)能够记录超过 70 分钟的数字化的、高质量的音乐节目。CD 的出现与迅速发展提醒了人们用它来记录计算机程序与数据。

用来记录计算机数据的光盘与记录音乐的光盘有不同的技术要求。首先，音乐的播放通常是顺序进行的，即从头至尾地播放；当需要从这个曲子跳到另一个时，跳跃的间隔也比较大。计算机数据的读取则不同，它要求可以从光盘的任何一点取出数据，这通常称为**随机访问**(Random Access)。其次，个别数据发生错误将降低音乐播放的质量，但是 CD 的误码率在不高于 10^{-8} 时，人们并不容易察觉到播放质量的降低。而对于计算机数据而言，这个错误率是不能容忍的。

随着光盘技术的发展，随机访问问题的解决，并且能够将误码率降低至10^{-12}，于1984年出现了记录计算机数据的CD-ROM(Read Only Memery)。最初1张CD-ROM的存储容量为660～1080 MB，读取速率是150 kB/s，寻道时间(即找到文件的起始位置的时间)为几百毫秒。至此，光盘的容量已经满足存储一个电影节目(1小时左右)的要求，而读出速率已经足够支持实时地提取已压缩的活动图像数据流的需要(1.2 Mb/s)，这就为多媒体技术的诞生提供了另一个必要条件。

差不多与CD-ROM迅速得到广泛应用的同时，只写一次光盘(Write Once)研究成功。这种12英寸的光盘最初能记录1 GB的数据，后来其容量也迅速扩展到5 GB和10 GB。

近年来可读/写光盘和计算机硬盘的容量迅速增大，并出现了磁盘阵列、光盘阵列等大型数据存储设备，为多媒体技术的实际应用和全面发展提供了充分的条件。

1.4 多媒体系统的基本类型及相关业务

多媒体计算机是多媒体技术的最直接、最简单的表现形式。因其本身具有存储、运算、处理和显示的能力，具有独立的功能，如动画显示、视频播放等，因此，多媒体计算机一出现便立即在家庭、教育和娱乐方面得到广泛的应用。但是，多媒体技术真正的意义在于与网络的结合，在于通过网络(局域网和广域网)为用户以多媒体的方式提供信息服务。

基本的多媒体系统除了以多媒体计算机为基础的独立(Stand-Alone)商亭式系统之外，通过网络提供业务的系统可以分为两大类：一类是人与人之间交互的系统，如多媒体会议与协同工作、多媒体即时通信等；另一类是人机交互的系统，如多媒体信息检索与查询、点播电视等，本节中将分别对这些系统及其技术特点进行介绍，而其中所涉及的关键技术将在以后的章节中加以讨论。

1.4.1 独立商亭式系统

凡是以一台多媒体计算机为核心的应用系统，例如商场的导购系统、展览馆的导游系统等，我们都称为独立商亭式系统。在这类系统中，除了各种媒体的采集、表示、压缩存储和解压缩播放之外，如何组织素材，并运用多媒体手段将信息有效地、具有感染力(或艺术性)和方便地提供给用户是制作应用软件时应考虑的重要问题。这里涉及的不仅有技术、有艺术，甚至还有社会、心理学等方面的问题。**多媒体制作软件**(如Authoring Tool、Authorware等)，或者原有操作系统的多媒体扩展(如Video for Windows)，是为制作应用软件而提供开发环境的软件。它不仅向应用程序的开发者提供多媒体输入/输出设备的接口，更重要的是，还提供建立媒体数据之间的空间布局和播放时间顺序等关系的手段。因此，开发优秀的制作软件本身远比开发应用软件困难。

在这类系统中，操作系统的实时性是值得重视的另一个问题。在嵌入式系统或工业控制机中常常涉及实时操作系统，在那里强调的是对事件中断的实时响应。而在多媒体系统中，由于视频和音频数据需要在一定时间约束条件下(如每秒25幅图像)连续不断地送到输出设备上供用户聆听和观看，因此这里操作系统的实时性强调的是，处理这些有时间要求的连续媒体流的能力。

提供更友好的人—机接口是商亭式系统技术发展的一个方向。除了使用键盘和鼠标，触摸式输入也很普遍，人们还试图通过声音、手势，甚至表情等多种模态的接口对系统进行控制，从而构成更人性化的多媒体交互环境。

除了独立商亭式系统外，下面将要介绍的4类系统都是在多媒体终端与终端之间、终端与应用服务器之间有网络相连接的**多媒体通信系统**。

1.4.2 多媒体信息检索与查询

通过因特网进行信息查询已是当前十分普及的应用。多媒体信息检索与查询MIS(Multimedia

Information Service)系统除可以根据关键字等对文本资料进行查询之外，也同时具有对活动图像和声音的查询能力。从通信方式而言，MIS 是点对点(信息中心对一个用户)，或一点对多点(信息中心对多个用户)的双向非对称系统。从用户到信息源只传送查询命令，要求的传输带宽较小，而从信息源传送到用户的信息则是大量的、宽带的[见图 1-1(b)]。

MIS 所涉及的两个重要技术问题是：① 如何向用户提供丰富的信息和如何让用户快速、有效地查询与浏览这些信息；② 如何合理、有效地组织多媒体数据的存储和检索。

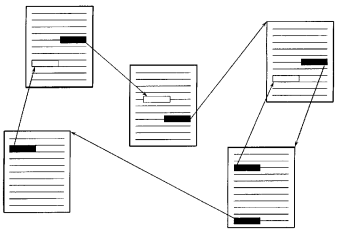


图 1-2　超文本文件结构

为了对第一个问题有所认识，首先让我们回顾一下人人都熟悉的读书过程。对于阅读一本小说来说，人们通常是从头至尾逐页阅读的，或者说是按顺序阅读的。但在有些情况下，特别是在技术或社会科学领域，在阅读某本书的过程中，经常需要从另一本书或论文查找某个论点，或者说，在几本书之间需要交叉参考的情况常常发生。图 1-2表示出用电子的方法来实现交叉参考的情况，这实际上已经是大家在因特网的查询中十分熟悉的过程：用鼠标点击黑框所标的地方，就会显示出箭头所示的有关参考信息，看完该信息后可以回到原来的页面，或者再进入其他页面 …… 箭头指向的页面(信息单元)可能与原来的页面在同一个文件中，也可能在其他文件里。这种信息的非顺序(或称为**非线性**)的组织结构称为**超文本**(Hypertext)，超文本中信息单元之间的链接称为**超链**(Hyper Link)。当上述信息组织方式不仅用于文本，还包括其他媒体数据、特别是音频和视频数据时则称为**超媒体**(Hypermedia)。超文本和超媒体这两个词在很多文献中也常常被混用。

超媒体为用户提供了一种在文件内部和文件之间迅速查找和浏览多媒体信息的方法，但是人们希望在更大的范围内迅速、有效地获取信息，这就不能不提到近年来推动因特网突飞猛进发展的**WWW 技术。**WWW(World-Wide Web)最初是 1989 年在日内瓦 CERN 启动的一个研究项目的名称，由于它的巨大成功，现在 WWW 已经意味着在超媒体原理下发展起来的一系列概念和通信协议。**Web** 这个词也代表了世界范围内由因特网相互连接起来的众多的信息服务器所构成的巨大的数字化的信息空间，也有的学者将之称为**超空间**(Hyperspace)。

WWW 的基本思想和它所解决的问题主要体现在如下几个方面：

(1) 在超空间中没有一个统一的管理者。任何人都可以创建超文本文件、将其与其他文件链接，并放入超空间中去。标准的超文本文件采用 **HTML**(Hyper Text Markup Language)格式。

(2) 定义了一种在超空间中寻找所需要的文件的机制，称为**统一资源定位器** URL(Universal Resource Locator)。通过 URL 可以知道每个文件处于哪一台机器，叫什么名字，以及以何种机制可以将该文件传输到需要链接它的地方去。

(3) 具有一个统一的、简单的用户界面，无论查询到的信息来自本机，还是来自远方的服务器，用户从界面上看起来都是一样的。实现 WWW 用户端功能的软件称为**浏览器**(Browser)。通过浏览器不仅能够调取 HTML 格式的文件，还可以调取以任何形式存储在已有的数据库、或信息库中的信息(虽然此时不具备超链接功能)。

以上 3 个问题的解决，使得世界上使用不同硬件和软件的分离的信息系统，通过因特网构成了一个庞大的统一的信息系统，从而为用户打开了通往一个大得难以想象的信息库的大门。这正是 WWW 取得巨大成功的原因。为了使用户不至于面对浩瀚的信息而不知所措，人们又进一步设计了帮助用户过滤掉无用信息、尽快找到所需要的信息的专门软件，这就是所谓的**搜索引擎。**

随着声音和活动图像等实时信息的逐步增加，因特网正在演变成世界范围内最大的 MIS 系统。由于以上所介绍的如何向用户有效地提供和查找信息的技术首先是从文字信息查询与检索发展起来的，关于这些技术的书籍已经很多，所以本书将不准备进一步讨论这方面的内容。

MIS 系统涉及的第二个重要技术问题是多媒体数据的存储和检索。与存储传统的数据不同，多媒体数据需要有适当的数据结构，以表达不同媒体数据之间在空间上与时间上的相互关系；对不同媒体要有合理的存储方式；对于数据量大而在时间上又有严格要求的音频和视频数据流，要有实时的提取算法；当数据库是分布式时，要能够将处在不同地域的服务器所提供的信息协调起来同步地提供给用户，等等。多媒体数据这种新型的数据给数据库的设计带来了一系列的新问题。目前的多媒体数据库一般是对通常的**关系数据库**进行扩充，或者采用**面向对象的数据库**来实现，数据量很大的视频则多以文件形式存储。多媒体数据库的成熟仍需要相当的时间。

此外，传统的、利用关键字或属性描述等来进行信息查询的方式，比较适用于文字信息，用来对声音、图像等多媒体信息的查询则有不方便之处。**基于内容的检索**是伴随着视频和音频查询而发展起来的新技术。利用这种技术，给出（或从查找对象中自动提取出）所要求的特征，例如图像中物体的形状、颜色等，就能找出具有同样、或类似特征的物体的图像来。更高级的查询方式则是给出“概念”或“事件”，如国旗、山脉、骑自行车的人等，找出具有同样概念或事件的图像或视频来。这种方式也称为**基于语义的检索**。基于内容和基于语义的检索涉及图像和视频的分析与理解、语义提取、模式识别与人工智能等，是当前多媒体领域中的一个重要研究方向。由于本书侧重于多媒体通信，因此将不准备讨论这方面的内容。

1.4.3 多媒体会议与协同工作

可视电话和会议电视是早在多媒体出现之前就已经存在的人与人之间进行通信的手段。**计算机支持的协同工作** CSCW（Computer Supported Co-operative Work）也是早在 20 世纪 80 年代初在计算机领域内提出的概念。它是指用来支持多个用户共同参与一件工作（如共同编辑文件、修改设计图等）的计算机系统及其相关的技术，但合作者之间不能“见面”与交谈。多媒体的出现为这两种交流形式提供了结合的基础，合作者既能看得见、听得到，又能一起处理事务，使他们真正像聚集在同一个房间里面对面地交流与工作。这种通信系统和业务称为**多媒体协同工作** MMC（Multimedia Collaboration）。多媒体远程医疗诊断系统、多媒体远程教育系统等都是融入了一定 MMC 功能的应用。

1. 会议室会议电视系统

这是传统的会议电视系统，其原理与广播电视系统类似，由电视摄像机对着主会场、主席等拍摄，通过电缆、光缆、微波或卫星信道送到分会场收看。如果要求主会场也能看到、听到分会场发言的情况，传输信道则是双向的，以将分会场的信号送到主会场。主会场（或者通信网的某个节点上）有信号切换设备，用来选取某一分会场的信号，并将该信号送至其他分会场；或者将几个分会场的信号综合起来，以分画面的形式送给各个会场。在有的系统中，主会场还可以对分会场摄像机的摄取方向等进行控制。图 1-3 是会议室会议电视系统会场的示意图。

这类系统的一个重要特点是，需要像电视台的演播室一样，对被拍摄的景物（人、黑板、会场的全景等）给以专门的照明（普通室内照明设施不能满足要求）。由于会议电视系统拍摄的景物没有什么剧烈的运动，主要是讲话人面部和形体的运动，而广播电视要传送包括诸如运动员的快速动作在内的高速运动的图像，所以会议电视的摄像机、信道设备等相对于广播电视所用设备而言比较简单，而且在同样的图像分辨率下，会议电视的数据率可以被压缩更大的倍数。另外，由于同样的原因，为了保证动作的连续，电视图像每秒钟需要传送 25 帧，而会议电视每秒传送 10～15 帧即可以被接受。而且，会议电视的情况与人们看电视不同，看电视是一种艺术欣赏，人们对图像的分辨率要求高，而会议则是在较长时间看一个相对固定的面貌，与会者在心理上对图像分辨率的要求大为降低。数据率为 384 kb/s 的系统所给出的图像质量已经可以令人足够满意了。

在会议室会议电视系统中，通过电子白板等辅助人机交互设备的使用，可以多媒体方式呈现、

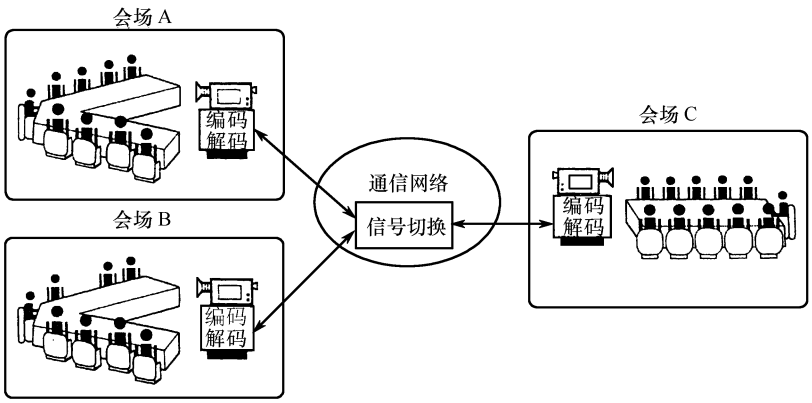


图 1-3　会议室会议电视系统

修改、记录和存储计算机中的文件，实现与会各方对信息的共享。

2. 桌面或手持终端会议电视系统

用计算机或手持智能设备取代会议室会议电视系统中的编解码设备和显示设备，是这类会议电视系统的基本特征。

在会议室会议电视系统中，摄像机不仅要拍摄讲话者还要能够对整个会场进行拍照，这要求摄像机有较大的视野和较高的灵敏度，因此其照明条件必须达到演播室的标准。而在桌面或手持系统中，摄像机只需要对准讲话的人，这不仅降低了照明要求，也降低了对摄像机的视野和灵敏度要求，摄像机大为简化。

在这类系统中，音、视频的处理与文字等其他媒体的处理被集成在一个系统中，这使得实现与会各方的信息共享和协同工作更为方便，有利于系统向多媒体协同工作的方向发展。

3. 多媒体协同工作

MMC 的最终目标是希望使身处异地的人们，能够像处于同一房间内面对面一样地交谈、协商工作，下面列举的是人们向着这一目标所正在作的努力。

教师从显示器屏幕的 3 个窗口分别看到在 3 个地方听课的学生，与在一个教室中面对全体学生的感觉是不一样的。利用计算机的图形功能可以生成类似真实图像的虚拟图像，例如具有天花板、窗户、灯具的教室，并将从 3 个地方传送来的学生的现场图像与计算机生成的虚拟教室图像结合在一起，构成一个全体学生在内的完整的教室全貌，将会给人以更真实的感觉。

在现实生活中举行会议时，某个与会者有时需要和邻座说一些不愿意让别人听到的悄悄话，或拿出一份文件与其小声商量；有时与会者要边讨论、边对一个文件或一份设计图纸同时进行修改，甚至需要共同操纵一台仪器进行实验；如此等等。在多媒体会议中，要实现类似现实生活中的这些行为要涉及许多技术问题。

显示器的屏幕是平面的，无论屏幕上显示的景象是多么的有立体感，人们仍然是身在其外，而不是身在其中。如何将**虚拟现实**(Virtual Reality)与协同工作结合起来，使人们在虚拟的三维环境之中协同工作是一个值得研究的课题。

人们会面时的第一个动作往往是一边握手，一边说“你好”。如果 MMC 终端可以用语言(不是键盘)输入、并配有机械手，可能使你感受到远方合作者向你握手问好的真实感觉。除了听觉和视觉之外，将其他的感觉，如触觉、嗅觉等结合进协同工作环境；或者将多媒体协同工作与机器人技术结合起来，使合作者能够共同进行除了屏幕上的工作(如编辑文件之类)以外的事情，这些都是研究者在探索的问题。

总之，多媒体协同工作将从各种不同的方面，向着能够使得被空间距离分开的人，在必要的时

候可以像已经聚在一起，有面对面地一起工作的条件与自我感觉的方向发展。但要真正达到这一目标，要走的路途还相当遥远。这里包括的不仅是技术问题，还有许多为社会学和心理学家们所感兴趣和值得研究的问题。

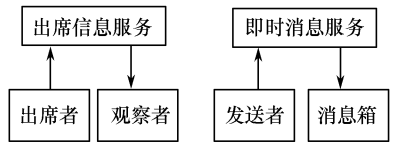
从通信的角度来看，MMC 系统是对通信系统要求最高的应用。它要求一点对多点，或者多点对多点的双向信息传输。所谓一点对多点，是指一个信号要同时向多个接收点传送。另外，在 MMC 系统中，声像信号是实时产生的，需要实时地压缩、传送，整个系统的时延要足够小，才能满足人们对话时自然应答的时间要求。在复杂的协同工作系统中，要实现“开小会”和进行共同操作等，还要能够随时建立、撤销某些专有信道。当涉及视、听之外的其他形式的传感器时，通信机制的复杂程度则会更高。

1.4.4 多媒体即时通信

即时通信系统更完整的表述是**出席与即时消息系统**(Present and instant messaging system，IMS)。它允许用户相互之间了解各自的状态和状态的改变，如在线、离线、繁忙、隐身等，并允许用户相互之间传递即时的短消息。第一个即时通信系统于 1996 年在以色列诞生。人们通过 IM 系统发送文本型的短消息，由于消息传送的即时性，对方可以立即给予回应，一来一往如同“聊天”；可以多个人一起聊，仿佛在一个聊天室，也可以两个人进行“私聊”，等等。由于这种交流方式的方便和快捷，IM 在世界范围内得到了迅速的发展，成为最流行的网络应用之一。现在，IM 从最初的个人聊天应用，逐步扩展到成为企业内部进行工作交流的有力工具，企业可以随时查看各部门在线人员情况，沟通各分支机构等。同时，IM 从原来支持简单的文本短消息交流，发展到加入文件传输、视/音频信息的即时传送，使聊天者相互可以看得见、听得着，等等。因此我们在本节标题中将它称为多媒体即时通信，它是一个极具潜力的业务。

加入了视、音频的 IM 系统从功能上讲与可视电话或会议系统类似，但实现方法并不相同。可视电话系统由通话双方通过呼叫协议直接建立双向的连接；而经典的 IM 系统采用客户端/服务器(C/S)结构，“聊天”双方的信息需要通过服务器进行中间转接。当传输视、音频信息时，由于数据量大，服务器中转可能引起响应的不及时，此时可以在“聊天”双方建立直接连接，但这个连接的建立通常也需要在服务器的帮助下完成。由于服务器是 IM 系统的核心，用户必须先登录服务器才能接受各种服务，因此服务器了解各用户的状态及状态的变化，从而能够向一个用户提供其他用户的状态信息，让他了解其他人的在线情况。这就是**“出席”**(present)服务。而在可视电话系统中，主叫方事先并不知道被叫方是否“出席”，他必须通过一定的通信协议呼叫对方(如振铃)，对方应答则接通；对方未出席，则不能接通。

如上所述，一个典型 IM 系统包含两种基本服务：**出席服务**和**即时消息服务**。图 1-4(a)为出席服务的基本框图。出席服务有两类客户，一类称为出席者(presentity)，另一类称为观察者(watcher)。出席者向出席服务提供自己的出席信息。观察者可以定期或不定期地向出席服务请求得到某些出席者的当前出席信息；也可以订阅(subscribe)出席信息，此时出席服务会在出席者的出席信息发生变动时主动告知订阅者。



(a) 出席服务　(b) 即时消息服务

图 1-4　即时通信

图 1-4(b)为即时消息服务的基本框图，其中发送消息的一方称为发送者，接收消息的一方称为即时消息箱。发送者向即时消息服务发送消息，消息中包含目的即时消息箱的地址；即时消息服务则根据目的地址向该即时消息箱转发消息。当用户之间需要交换视、音频消息时，发送者从服务器获得接收方的地址和状态信息，并通过一定的协议在服务器的帮助下建立起二者之间的直接连接。然后在此连接上视、音频数据可以采用与可视电话和会议系统中类似的方式进行传输。

1.4.5 点播电视(VOD)

多媒体计算机出现以后，由于它具有以交互操作的方式调取包括伴音在内的活动图像的功能，立即导致了这样的构思：在电视节目中心，将节目以压缩后的数据形式存入视频节目库；用户在家里可以按照菜单调取任何一套节目，或者一套节目中的任何一段，并能实现录像机的功能，即快进、快退、重放、慢动作以及播放静止画面等，这种系统与业务便是**点播电视**(Video on Demand，VOD)。

从使用功能上讲，VOD 与多媒体信息检索与查询系统是类似的，但是二者的业务特点却有很大不同，因而技术的侧重点也有所不同。在多媒体信息查询系统中，信息的主要部分通常是通过文字、图片表达的，数据量较小，其中需要显示的视频图像片断一般时间不长。此外，在多媒体信息查询系统中，用户的查询时间和查询内容不像人们点播电视节目那样集中。在 VOD 系统中，由信息中心送出的是数据率很高的电视节目，其中具有录像质量的信号的数据率为 1.5 Mb/s，电视质量的信号则达 2～4 Mb/s。电视节目的长度是以小时计算的，在这几个小时中，每一秒钟内都需要传送几兆比特的数据，才能使用户正常、不中断地收看节目。另外，人们看电视的时间比较集中，晚上看电视的用户比白天的多，晚饭后(即黄金时间)看的人则更多；看的节目也比较集中，看好片、新片的人比看劣片、旧片的人多。不难想象，要在同一时间内向数百、数千、甚至更多的用户提供内容不同，而又连续不间断的高速数据流，对设备和网络的要求是很高的。

图 1-5 是一个 VOD 系统示意图。图中的磁盘中存储着大量电视节目；负责按用户提出的要求向用户传送节目数据流的设备称为**视频服务器**，或流服务器。用户终端则用来接收节目数据、并将其解码、还原成模拟电视信号进行显示，同时，还负责将用户的查询命令发送到上行(即由用户向信息中心的方向)线路上。用户终端可以电视机、计算机或手持智能设备为平台。当使用不具备智能功能的电视机时，需附加机顶盒以完成与服务器的交互操作。

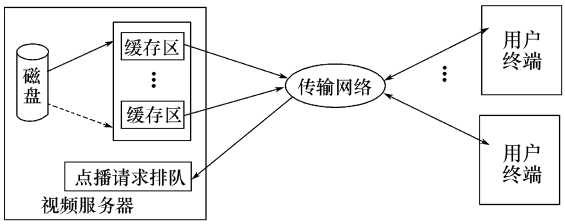


图 1-5　VOD 系统的简单示意图

现在让我们来考虑视频服务器如何为众多的用户服务。必须注意，在任何瞬间，服务器只能从一个磁盘上提取一个节目的数据。如图 1-5 所示，假设我们在服务器中为每一个用户设立一个缓存区。服务器首先将第一个用户需要的节目全部从这个磁盘中提取出来，放入该用户的缓存区(内存)，然后再为第二个用户提取数据 ……由于前一个用户所需要的全部数据都已存放在缓存区中，所以在服务器为后一个用户提取数据时，并不影响在前一个用户的缓存区内的数据持续不断地向该用户输出。这是实现视频服务器功能的一种最简单的设想。但是此方法需要容量巨大的内存作为用户的缓存区；同时，稍后提出请求的用户要等到将前面的用户在未来数小时内(即整个节目)所要用到的数据都提取完之后，才能得到服务，等待时间太长。

实际视频服务器所采用的一种典型方案如图 1-6 所示。在这个方案中，一个节目被分成若干段，每一段分别存储在一个磁盘中(多个磁盘构成一个阵列)。当服务器从第一个磁盘为用户 1 取出一段数据送至用户 1 的数据流缓存区中后，用户 1 就可以开始获得数据、观看节目了。这时服务器则接着从第二个磁盘中为用户 1 取第二段数据，在用户 1 的缓存区中的数据被用空之前，将第二段数据补充进去，使送至用户 1 的数据流不至中断；与此同时服务器又从第一个磁盘中为用户 2 取出第一段数据等等。如果从磁盘中提取数据的速率超过向用户传送数据的速率，磁盘数 m 可以

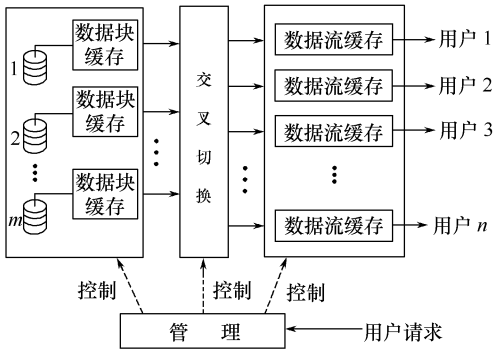


图 1-6 视频服务器

小于用户数 n。从磁盘中提取数据的速率越快，可以服务的用户数越多。图中的交叉切换模块代表对各个数据流进行调度和控制的硬件与软件。

可以看到，与图 1-5 所示的方案相比，在内存的占用量和用户等待时间方面这种方案要优越得多。但是由此我们也看到，这里的视频服务器，其硬件和软件比多媒体信息查询系统中的 Web 服务器要复杂得多。再设想一下，若干个用户可能在相近的时间，甚至在只差几秒钟的情况下点播同一个节目，并且某些用户要进行快进、快退等录像机式的操作等，要满足这些要求将进一步增加服务器数据提取、调度和控制的复杂性。

在通信网络方面，相比于其他多媒体应用，如信息查询、会议电视等，VOD 是对带宽要求最高的应用。在用户端，对于固定网络而言，必须采用电话线或同轴电缆的宽带接入线路，或者光缆直接铺设到用户；而对于移动网络，则只能在 3G 或更先进的网络上实现。当存储节目的信息中心与用户在地理位置上相距很远时，要在远程网络上长时间地传送众多的持续不断且速率很高的数据流，对网络是一个沉重的负担。如何有效地进行这类通信，在后面的章节中我们将会看到，人们已经提出了一些解决方案。

1.5 三网融合及相应的业务

如本章开始所述，计算机、通信和广播电视原本是三个独立的技术领域，它们之间的相互渗透和融合产生了多媒体技术；而多媒体技术出现以来，对多媒体业务的需求又进一步促进了这三个领域的融合。特别是近十年来，在“三网融合”方面，即提供电话服务的固定和移动通信网、提供数据服务的计算机网和提供电视服务的广播电视网之间的融合，取得了明显的进展，从而为不同业务的融合和新业务的产生展示了广阔的前景。本节简要回顾有关的进程。

1.5.1 网络的融合

20 世纪 90 年代关于 ATM 和 IP 之争在三网融合的发展过程中占有至关重要的地位。ATM 是国际电联为宽带综合业务数字网(B-ISDN)所规定的传输模式，而 B-ISDN 则是当时人们期望的下一代通信网络，未来各种宽带业务都希望通过 B-ISDN 这个统一的传输平台来提供。B-ISDN 虽然具有较为先进的设计，并被期待具有理想的性能，但是它在许多方面，如交换设备、终端设备、传输模式等，与现有的通信系统有较大的区别。对于庞大的现存系统而言，“废旧立新”遭遇到强大的社会和经济阻力，因此 B-ISDN 并未得到预期的发展。IP 是 20 世纪 80 年代出现的用于计算机网络的一组协议，它比较好地解决了采用不同传输协议的网络之间相互连接问题。通过 IP，不同的网络之间能够进行数据交换，并能够向用户提供一致的通信服务。正是借助于 IP，分散在不同地

方的不同网络才得以互相连接构成了一个世界范围内的统一的网络——互联网。也正是由于IP这种开放性，即允许物理层和数据链路层采用不同的传输协议，并允许不同的应用在它之上工作，使得它在与ATM之争中获胜，并已成为通信网和计算机网融合中网络层的事实上的标准。对于通信行业的从业人员来说，十几年前IP还是一个生疏的计算机领域的名词，而今天却是一个必须掌握的基本概念了。

基于IP的网络，如互联网，初期主要用于传送计算机数据，其传输方式和性能不能满足传送数据量大、实时性要求高的音频和视频的需要。为了能在IP网上提供电话、电视和各种多媒体业务，各国生产和运营企业，以及研究和工程技术人员进行了大量的工作，这些工作的进展主要体现在两个方面：一是增加核心网和接入网的带宽，为各种业务，特别是宽带多媒体业务提供充足的资源；二是设计有效的资源分配和管理机制，在有限的资源下，为不同的媒体提供不同的服务保障，例如保障数据传输的可靠性和保障音频、视频传输的实时性等。后者常称为**服务质量**(Qualify of Service，**QoS**)保障机制。近年来，对于现已存在的网络和新出现网络的QoS机制的研究受到了广泛的重视，这些工作为在融合的(IP)网络上不同媒体数据，特别是音、视频数据的传输改善了条件。

同时在核心网方面，一个值得关注的技术趋势是网络控制和数据传输功能的分离。在传统的通信网中，电话的呼叫控制、路由选择和数据转发都集中在程控交换机中完成。而数据在分组交换的网络上传输时，则需要采用另一种类型的节点设备(如路由器)来完成呼叫、路由和数据转发等功能。近年出现的技术，如**软交换**、通用多协议标记交换(**GMPLS**)等，在网络节点处将控制和传输功能分离，使这两个功能得以分别独立地演变和发展，从而为核心网络的融合提供了极大的灵活性。新的传输技术发展过程中出现的新交换方式，如光网络中的波长交换和光纤端口交换等，可以与传统的时分复用交换和分组交换一起，在节点由统一的控制模块控制，从而建立起不同类型线路之间的连接通路。

在接入网方面，原本接入用户家庭的传送电话的电话线和传送有线电视节目的同轴电缆，通过xDSL和电缆调制/解调等技术已经能够同时支持电话、电视、数据和各种多媒体业务；而更令人瞩目的是无线接入方式的发展。以电话为主要业务的第二代(2G)移动电话网已经进入第三代(3G)，在保持原有良好移动性的同时，扩展到更大的带宽，并从传统的电话通信模式转向在IP的框架下工作。在另一方面，由计算机网络发展而来的无线局域网(WLAN)和无线城域网(WMAX)在具有较大带宽的基础上，正在向支持更好的移动性的方向发展。可以期望这3种无线网络将逐渐演变和融合成为宽带化、移动化和全IP化的网络，从而与各种固定接入网一道，为多媒体业务提供“无处不在”的接入服务。

21世纪初由移动通信业界联盟3GPP所提出、后来为包括国际电联在内的其他国际组织所接受的**IP多媒体子系统(IMS)**，是网络融合发展过程中的又一重要成果。目前IMS已由无线网络扩展到有线网络，并成为下一代网络框架的核心。在传统的电话业务中，通过发送方拨叫接收方的电话号码，网络可以为任意两个终端建立一对连接；而在IP网上，这一至关重要的连接能力只能相对隔离地在单个服务提供商内部提供。因此，我们需要一个全球的系统，以使得各种类型终端在IP之上的应用都可以相互建立起对等的通信，IMS就是这样一个系统。IMS除会话管理之外，还涉及完成服务的其他功能，如注册、安全、计费、网络控制、漫游和QoS等；同时它与现存的语音和数据网络可以实现互通。

从以上几个方面的分析，我们可以清晰地看到如下的趋势：电话、移动电话、因特网业务和电视原来是由电话网、移动电话网、计算机(IP)网和广播电视网4个独立的网络分别传送的。现在由于业务层、控制层和连接层的分离，不同类型的传输技术和接入技术可以统一在一个通信框架下；同时，各种业务均建立在IP之上，通过IMS进行控制，这就构成了一个能够提供电话、电视、因特网业务及其他现有的和未来可能出现的各种多媒体业务的统一的平台，使得“在任何时间、任意地方、

通过任何形式的终端接受任何类型服务”的理想具备了实现的可能性。

1.5.2 多重服务与业务融合

在网络融合的演进过程中，向用户提供的业务也在相应地发生着变化。移动电话网上的数据业务——短信，在世界范围内获得了巨大的成功。利用 IP 进行电话传输的技术——VoIP 已为人们所熟悉；而利用 IP 提供（直播和点播）电视业务的 IPTV 正在迅速发展之中。通过手机不仅可以打电话、送短信、上网，还可以接收图片和音乐，甚至可以看“手机电视”。人们通过数字电视的机顶盒和电视机不仅可以看电视、点播电视节目，还可以上网和进行电子交易等；而通过计算机终端则不仅可以上网、下载文件，还可以打可视电话、收看电视节目；等等。原来针对某种单一业务的终端，如，电视机、计算机和手机，逐渐变为“多功能”的多媒体终端。1.4 节中介绍的几种基本多媒体业务互相渗透，行业的界线逐渐模糊，电话、电视和数据不再必须由分离的专用网络和不同行业的运营商所提供。

随着三网融合的发展，在统一的网络框架上由一个运营商就可以向用户提供电话、电视、宽带数据及其他多媒体业务，这就是**三重服务**（Tripe play）、或多重服务的概念。在通过 IMS 实现如前所述的网络融合架构中，运营商通过统一的应用层和控制层向不同接入网上的用户不但可以提供 1.4 节所述的多种业务，而且用户还可能在一项通信过程中，使用不同的终端设备、跨越不同的接入网、获得不中断（无缝）的服务。例如，一个用户在飞机场用多模手持设备通过 WLAN 给正在办公室内的同事打可视电话，所谓多模，指其终端具有接入多种接入网的接口。该用户出飞机场后在汽车上改至 3G 网络通话，后来他们觉得需要邀请另一个同事加入讨论，该同事接收邀请在办公室通过计算机参与电视会议，会议期间他们还可以相互发送信息、观看视频或共同编辑一个文件等。或者，一个用户在出租车内用手持设备观看某场直播的球赛，当他回到家后，可以转至大屏幕电视上继续观看。可以期待，随着网络融合的进展，更新和更吸引人的多媒体业务也会随之产生。

习 题 一

1. 请论述什么是多媒体和多媒体技术的基本特征。
2. 多媒体系统有哪几种主要的类型？每一种类型中最关键的技术有哪些？
3. 三网融合已经取得什么样的进展？还存在哪些困难？
4. 试建议几种在网络融合基础上可以提供的新业务。

参 考 文 献

[1] G. R. Wichman, “Software without Border,” Sun World, Vol. 4, No. 12, 1991

[2] N. D. Georganas, “Multimedia Communication,” IEEE J. SAC, Vol. 8, Vo. 4, 1990

[3] D. Minoli and R. Keinath, Distributed Multimedia through Broadband Communications Services, Boston; Artech House, 1994

[4] A. C. Luther, Digital Video in the PC Environment, New York: McGraw-Hill, 1991

第 2 章　视觉特性和彩色电视信号

2.1　人的视觉特性

2.1.1　图像对比度与视觉的对比度灵敏度特性

1. 图像的对比度

对比度表示图像相邻区域或相邻点之间的亮度差别。对比度 C 由下式定义：

$$C = \frac{I_{\max}}{I_{\min}} \tag{2-1}$$

式中，$I_{\max}$、$I_{\min}$分别代表图像中的最大和最小亮度。

在自然景物中，对比度经常可以达到 200∶1，甚至更高。电视机和显示器只有给出类似的对比度，电视上的景物才有自然景物那么明亮和层次丰富。

2. 视觉的对比度灵敏度特性

在给定的某个亮度环境下，人眼刚好（以 50%的概率）能够区分两个相邻区域的亮度差别所需要的最低对比度，称为**临界对比度**，或称为**视觉阈**。在研究数据压缩技术时，人们关心人眼是否能够察觉到压缩所引入的图像（对比度）失真，因而对视觉阈的研究就是十分必要的了。临界对比度的倒数，称为**对比度灵敏度**。

视觉阈的大小与观察条件（如周围环境的亮度、邻近区域亮度的变化等）有关。假设我们考虑图 2-1(a)所示的情况，其中环境亮度为 L_S，图的中间有一个张角为 1.5°、亮度为 L_B 的环，环内包围着一个亮度为 L 的小区域。调节 L 的大小使其刚好能被觉察到与 L_B 有所不同，则 $L-L_B=\Delta L$ 为视觉阈。在不同的 L_S 下，经实验得到 ΔL 随 L_B 变化的曲线如图 2-1(b)所示。在 $L_S=L_B$ 的情况下，$\Delta L/L_B$ 接近于常数，这称为**韦伯**(Weber)**定律**。ΔL 与 L_B 成正比意味着，人眼区分图像亮度差别的灵敏度与它附近区域的背景亮度（平均亮度）有关，背景亮度越高，灵敏度越低。

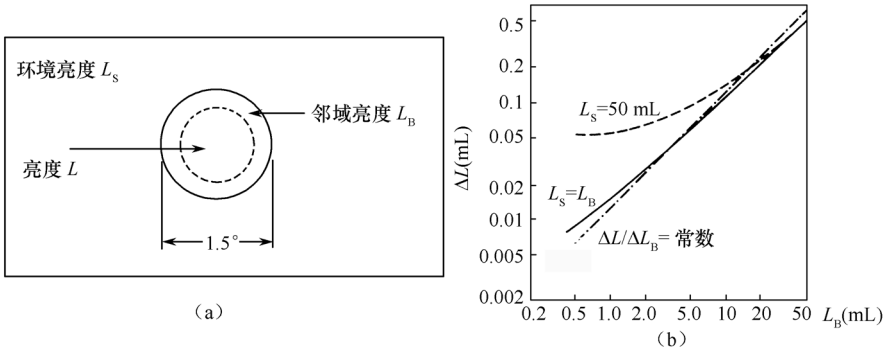


图 2-1　视觉阈的测量（背景亮度固定）[1]

3. 空间域的掩蔽效应

视觉阈的大小不仅与邻近区域的平均亮度有关，还与邻近区域的亮度在空间上的变化（不均匀性）有关。假设将一个光点放在亮度不均匀的背景上，通过改变光点的亮度测试此时的视觉阈，人们发现，背景亮度变化越剧烈，视觉阈越高，即人眼的对比度灵敏度越低。这种现象称为视觉的**掩蔽效应**(Masking)。

2.1.2 空间频率与视觉的空间频率响应

1. 空间频率

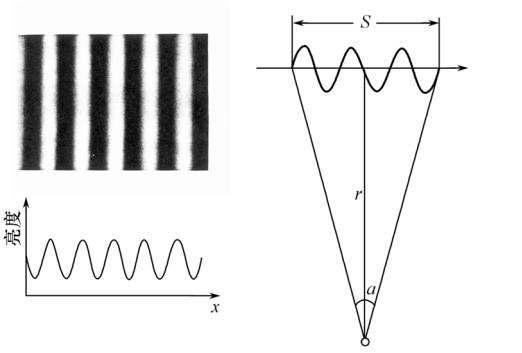
人们所熟悉的**时间频率**是用单位时间内某物理量(如交变的电流、电压、波动或机械振动等)周期性变化的次数来定义的,单位为周/秒。类似地,**空间频率**的定义是:物理量(如图像的亮度)在单位空间距离内周期性变化的次数,即

$$f_x = \frac{\mathrm{d}\varphi(x)}{\mathrm{d}x} \text{(周 / 米)} \tag{2-2}$$

式中,f_x 表示亮度信号在 x 方向的空间频率,x 是空间距离变量,$\varphi(x)$ 表示亮度信号沿 x 方向的相位变化。

以图 2-2(a)所示的正弦光栅为例,如果光栅亮度在 1 厘米内变化 10 次,那么它的空间频率为 $f_x=1000$ 周/米。

人们通常有这样的经验,从不同距离上观察空间频率相同的正弦光栅,感觉光栅亮度变化的密集程度是不同的。因此在涉及观察者时,需要将空间频率用每度多少周表示,这里的度是几何角度的单位。这样表示的空间频率可以理解为从某一观察点来看,亮度信号在单位角度内周期性变化的次数,即



(a) 正弦光栅　　(b) 空间频率的两种单位之间的转换

图 2-2　空间频率的单位

$$f_x = \frac{\mathrm{d}\varphi(\alpha)}{\mathrm{d}\alpha} \qquad \text{(周 / 度)} \tag{2-3}$$

式中,α 为图 2-2(b)中所表示的角度,$\varphi(\alpha)$ 代表亮度信号在角度 α 内的相位变化。两种空间频率单位间的转换可以通过图 2-2(b)来推算。假设图中 $S \ll r$,则存在如下的近似关系:

$$S = 2r\tan\frac{\alpha}{2} \approx 2r\frac{\alpha}{2} = r\alpha \tag{2-4}$$

式中,α 的单位为弧度。因为信号在 S 长度内变化的周期数等于在 α 角内变化的周期数,故可得下式:

$$f_{xm}S = f_{xd}\,\alpha\,\frac{180^\circ}{\pi} \tag{2-5}$$

式中,f_x 的下角标 m 和 d 分别用来区别以长度为单位和以角度为单位的空间频率。将(2-4)式代入(2-5)式,得到 f_{xd} 和 f_{xm} 之间的转换公式

$$f_{xd} = \frac{\pi}{180^\circ} r f_{xm} \text{(周 / 度)} \tag{2-6}$$

二维图像的空间频率谱可以用二维的傅氏积分来表示,即

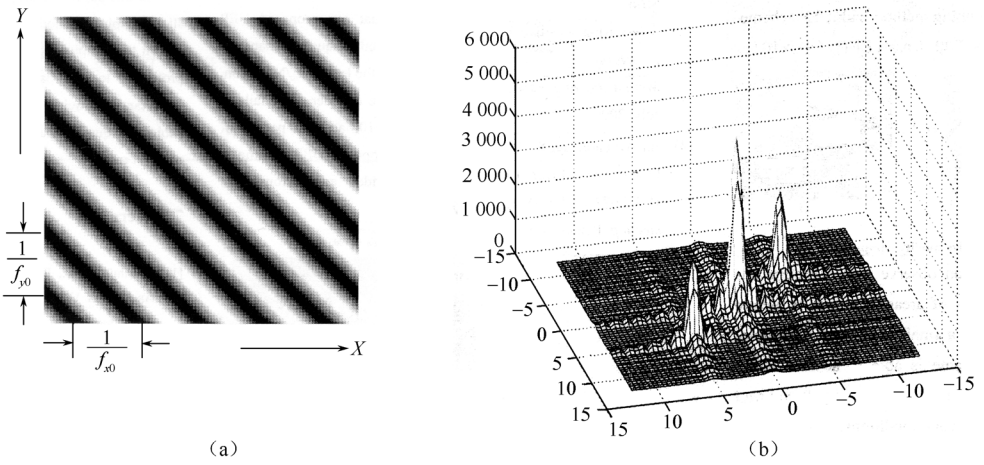
$$F(f_x, f_y) = \int_{-\infty}^{\infty}\int_{-\infty}^{\infty} L(x,\ y)\mathrm{e}^{-\mathrm{j}2\pi(f_x x + f_y y)}\,\mathrm{d}x\mathrm{d}y \tag{2-7}$$

式中,$L(x,y)$ 为亮度在 $x-y$ 平面上的分布函数,x、y 为图像的平面坐标,f_x、f_y 分别是在 x、y 方向上的空间频率。

对于图 2-3(a)所示的正弦光栅,其亮度函数为

$$L(x,y) = \frac{1}{2}[1 + \cos 2\pi(f_{x0}x + f_{y0}y)] \qquad (f_{x0} > 0 \quad f_{y0} > 0) \tag{2-8}$$

根据(2-7)式得到它的频谱如图 2-3(b)所示,其中包含直流分量和 $f_x = f_{x0}$、$f_y = f_{y0}$ 的频率分量。与一维的频谱分析相类似,任何一个复杂的亮度函数(图像),都可以通过傅氏分析将其分解成为一系列不同频率分量(正弦光栅)之和。



(a)　　(b)

图 2-3　一个正弦光栅和它的空间频率谱

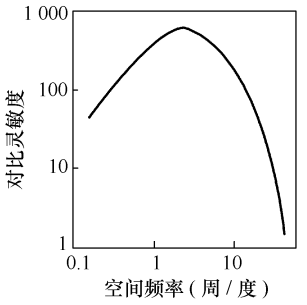


图 2-4　视觉的对比度灵敏度响应
(正弦光栅的亮度为 600 坎德拉/米^2)[1]

2. 视觉的空间频率响应

人的视觉系统基本上可以认为是一个线性系统。图 2-4给出了视觉对不同空间频率的正弦光栅的响应[1]，它表示了该线性系统的频率域特性。从图可以看出，当空间频率在 3～4.5 周/度时，视觉的对比度灵敏度最高，即人眼对这些空间频率的分辨能力最强。由于眼睛的光学孔径大小的限制和视网膜上光敏细胞分布的密度不够等原因，对于空间频率高于 50～60 周/度的正弦光栅，人眼就很难分辨了。视觉系统频率响应在低端的下降，则需要用眼睛的横向抑制效应来解释[2]。

2.1.3　视觉的时间域响应

1. 视觉惰性与闪烁

人眼的视觉是有惰性的，这种惰性现象也称为视像的暂留。就是说，当眼前实际的景物 已经消失后，所看到的影像却不立即消失。如果让观察者观察按时间顺序重复的亮度脉冲(如黑暗中不断开、关的手电筒)，当脉冲重复频率不够高时，人眼就有一亮一暗的感觉，称为**闪烁**。如果重复频率足够高，闪烁感觉消失，看到的则是一个恒定的亮点。闪烁感觉刚好消失时的重复频率叫做**临界闪烁频率**。脉冲的亮度越高，临界闪烁频率也相应地增高。

视觉惰性现象已被人们巧妙地运用到电影和电视当中，使得本来在时间上不连续的图像，给人以真实的、连续的感觉。在通常的电影银幕亮度下，人眼的临界闪烁频率约为 46 Hz。所以电影中，普遍采用每秒钟向银幕上投射 24 幅画面的标准，而在每幅画面停留的时间内，用一个机械遮光阀将投射光遮挡一次，得到每秒 48 次的重复频率，使观众产生亮度是连续的、不闪烁的感觉。人们也曾做过用遮光阀将每幅画遮挡两次的实验，这时可以在不产生闪烁感觉的前提下将每秒钟投影的画面幅数减少到 16，从而能够进一步缩短电影复制所需的胶卷的长度。但是，每秒钟投影 16 幅画面时，对于速度稍高的运动物体，由于前一幅画面和后一幅画面中的物体在空间位置上的差别过大，会产生像动画片那样的动作不连续的感觉。在 2.2.2 节中我们将会看到，类似的思想也在电视中得到了应用。

实验表明[3]，人眼在高亮度下对闪烁的敏感程度高于在低亮度的情况。对于今天的高亮度的显像管而言，临界闪烁频率可能达到 60～70 Hz。

2. 运动的连续性

一般来说，要保持画面中物体运动的连续性，要求每秒钟摄取的画面数约为 25 帧左右，即帧率要求为 25 Hz；而临界闪烁频率则远高于这个频率。在传统的电视系统中由于整个通道中没有帧存储器，显示器上的图像必须由摄像机传送过来的画面所刷新，所以摄像机摄取图像的帧率和显示器显示图像的帧率必须相同，而且相互是同步的。在数字电视和多媒体系统中，在最终显示图像之前插入帧存储器是很简单的事，因此摄像机的帧率只要保证动作连续性的要求，而显示器可以从帧存储器中反复取得数据来刷新所显示的图像，以满足无闪烁感的要求。现在市面上出现的 100 Hz 的电视机，就是用这种办法把场频由 50 Hz 提高到 100 Hz。

3. 时间域的掩蔽效应

影响时间域掩蔽效应的因素比较复杂，对它的研究还处于初始阶段。这里仅介绍一些实验结果，这些结果可能在数据压缩方面有潜在的应用价值。实验表明[4]，当电视图像中相邻的画面变化剧烈（如场景切换）时，人眼的分辨力会突然剧烈下降，例如下降到原有分辨力的 1/10。也就是说，当新场景突然出现时，人基本上看不清新景物，在大约 0.5 秒之后，视力才会逐渐恢复到正常状态，很显然，在这 0.5 秒的时间内，传送分辨率很高的图像是没有必要的。研究者还发现[5]，当眼球跟着画面中的运动物体而转动时，人眼的分辨力要高于不跟着物体而转动的情况。而通常在看电视时，眼睛是很难跟踪运动中的物体的。

2.1.4 彩色的计量和彩色视觉

1. 彩色的定量表示

人们在生活中，用红、橙、黄、绿、青、蓝和紫等名词来描述彩色的大致范围。如果再进一步地细分，红色则有深红、浅红、大红、粉红等。即使这样细分，仍然不能把颜色表达得很准确。

根据德国科学家格拉兹曼所总结的法则[6]，任何一种彩色都可由另外的不多于三种的其他彩色按不同的比例合成。这意味着，如果选定了三种人所共知的标准基色（标准基色必须是独立的，即其中一种不能由其他两种产生），那么任何一种彩色，可以用合成这一彩色所需要的 3 种基色的数量来表示。例如，选择波长分别为 700 nm、546.1 nm 和 435.8 nm 的红、绿、蓝光作为基色，用不同比例的三基色光可以配出任何一种彩色。三种光的能量之和决定了合成光的**亮度**，而三种光强之间的比例关系决定了合成光的**色调**（颜色）和**饱和度**（颜色深浅）。一个任意光（A）和三基色光之间的关系可以写成下式

$$(\mathrm{A}) = r_a(\mathrm{R}) + g_a(\mathrm{G}) + b_a(\mathrm{B}) \tag{2-9}$$

式中带有括号的大写字母只代表某种光。例如（R）只代表红光，并不具有数量和量纲的含义，数量由它们各自的系数代表。

(2-9)式表明，在基色光（R）、（G）和（B）选定以后，任何一种彩色（A）都可以用三个相应的数 r_a、g_a 和 b_a 来表示。这事实上已经解决了用数学的方法严格地定义彩色的问题。但是在实际的应用中发现，这样的三个数有时相互之间在数量上可以相差几个数量级，以至于有的数值小到在进行色度计算时可以忽略，而它在光的合成中却起着明显的作用，又不能忽略。解决这一问题的办法，是用合成某种标准白光（如等能白光）所对应的三个系数值，分别作为三种基色光的 1 个计量单位。以此计量单位度量的任意彩色（A）的三个系数称为**三色系数**（Tristimulus）[7]，用 R，G，B 表示。

在 R、G、B 三种基色构成的颜色空间中，以三色系数为坐标，任何一种彩色都可以由这三个坐标值所确定的矢量来表示。矢量的幅值代表了彩色的亮度，矢量的方向代表了它的颜色信息（色调和饱和度）。图 2-5 中，坐标原点 O 是 R、G、和 B 都等于零的点，代表黑色。与 O 点相对的立方体的顶点的三个坐标值都等于 1，代表等能白色。从图看出，具有同样颜色、不同亮度的矢量都与虚线三角形相交于同一点，这说明虚线平面三角形内的一点，唯一地确定了一种颜色；三角形包围的

区域定义了在这种三色坐标下所有可能产生的颜色。

图 2-5 中虚线平面三角形中的一点唯一地确定一种彩色，说明彩色只需要两个（而非三个）变量就可以描述。我们将三色系数作如下的归一化：

$$r=\frac{R}{R+G+B}$$
$$g=\frac{G}{R+G+B} \tag{2-10}$$
$$b=\frac{B}{R+G+B}$$

则有 $r+g+b=1$。这样，一种彩色只需要两个变量，例如 r 和 g，就可以严格定义了。

由于用 r 和 g 来描述自然界中的彩色有计算上的不便之处，国际照明委员会（CIE）于 1931 年基于假想的三个基色 X,Y,Z 建立了一个标准色度系统。XYZ 系统中的三色系数可以由 RGB 三色系数经线性变换获得[7]。在 XYZ 系统中，以 x 和 y（对应于 RGB 系统中的 r 和 g）为坐标表示的色度图如图 2-6 所示。图中舌形的区域展示了人类可见的所有彩色，外侧边界是光谱（单色）光轨迹。一个多媒体或电视系统能够展示给观众的彩色由其显示设备所使用荧光粉的发光特性所决定。图 2-6 中实线和虚线三角形的三个顶点分别代表 PAL 和NTSC电视系统 R、G、B 荧光粉的色坐标，中间 E 代表等能白色的色坐标，三角形包围的面积代表该系统可以展示的彩色。

除了 RGB 和 XYZ 坐标系外，还有其他的彩色坐标系统，如在电视和多媒体系统中经常使用的 YUV（见 2.2.4 节），在艺术和计算机视觉领域广泛使用的 HSV、HSI 等。有兴趣的读者，可以参阅有关的文献[1,8]。

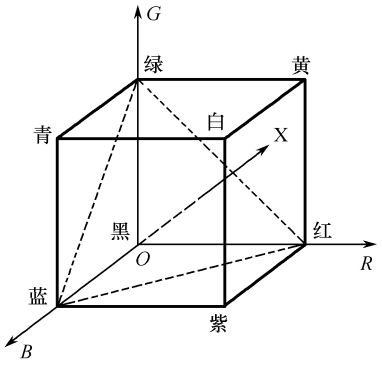


图 2-5　三维彩色矢量空间

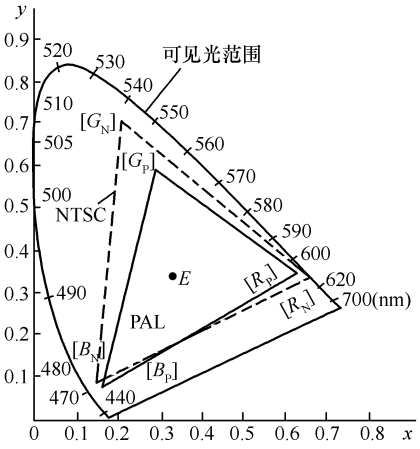


图 2-6　x-y 坐标色度图

2. 彩色视觉的空间频率响应

图 2-7 给出了视觉对彩色变化的频率响应。由图看出，人眼对亮度的分辨力要明显的比彩色的高。对间隔较密的黑白正弦光栅我们可能可以分辨清楚，而同样间距的蓝黄光栅，我们可能分不清，而只能看到一片绿。

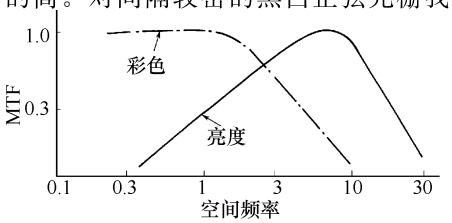


图 2-7　彩色视觉的空间频率响应

3. 彩色的掩蔽效应

在亮度变化剧烈的背景上，例如在黑白跳变的边沿上，人眼对色彩变化的敏感程度明显地降低。相类似地，在亮度变化剧烈的背景上，人眼对彩色信号的噪声（如彩色信号的量化噪声）也不易察觉。这些都体现了亮度信号对彩色

信号的掩蔽效应。

2.2 彩色电视信号

2.2.1 扫描——空间频率到时间频率的转换

在电视系统中，摄像管是通过电子束扫描（利用CCD摄取信号的过程与这一方式完全等效）将图像分解成与像素对应的随时间变化的电信号的。在接收端，显像管则以完全相同的方式，利用电子束从左到右、从上到下的扫描，将电视图像在屏幕上显示出来。

假定落到摄像管光电靶面上的图像是一幅正弦光栅，如图2-8(a)所示，其光栅亮度 L 沿水平方向 x 的分布是按图(b)所示的正弦规律变化的。在摄像管中，电子束从左到右扫描形成电信号。图(c)是与图(a)相对应的一个扫描行上的电信号。该信号称为视频信号或电视信号，其特点是它的下限频率可以接近于零频。图(b)和图(c)都是正弦波，但它们的纵坐标一个是亮度，另一个是信号电流，而横坐标则分别为水平距离 x 和时间 t。

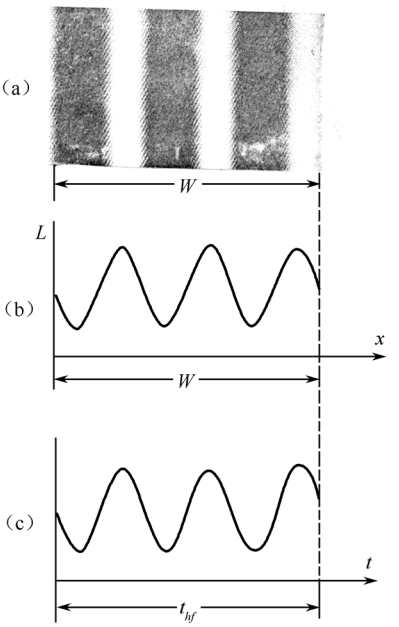


图2-8 正弦光栅对应的视频信号

假设 f_x（周/米）为正弦光栅的空间频率，W 为光栅的宽度，f_xW 则为电子束在一行扫描中所扫过的亮度波形的总周期数。电子束从左至右的扫描，称为**行正程扫描**；从右回到左端，则称为**行逆程**。摄像管电子束在逆程期间不拾取信号，因此，显像管在逆程期间也不呈现图像。设行正程对应的时间为 t_{hf}。对比图(b)与(c)可以看到，在时间 t_{hf} 内，电信号变化总周期数等于亮度信号在 W 内变化的总周期数，因此，电信号的时间频率 f 可以表示为

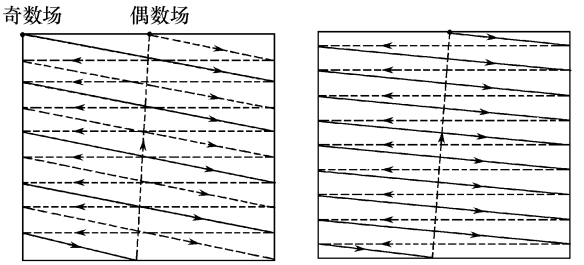
$$f = f_x \frac{W}{t_{hf}} \tag{2-11}$$

式中，W/t_{hf} 为电子束的正程扫描速度。(2-11)式说明了在将光像转换成电信号过程中的一个基本概念：当需要传送的图像细节 f_x 固定时，视频信号的频率 f 与扫描速度成正比；或者，在扫描速度固定时，信号频率 f 与要传送的细节 f_x 成正比。

2.2.2 隔行扫描与逐行扫描

在电影技术中，每秒钟向银幕上投影24幅画面，再将每一幅画面用遮光阀挡一次，从而得到48次的重复频率，将一个电影胶卷的长度降低了一倍。与此类似，在电视中采用了图2-9(a)所示的**隔行扫描**方式：第一场（称为奇数场或顶场）扫描第1、3、5等奇数行（实线所示），第二场（称为偶数场或底场）扫描2、4、6等偶数行（虚线所示）。两场合起来构成一幅画面，称为一**帧**。这样，每秒钟光栅闪烁的次数是50次，而实际显示的画面只有25幅，即场频为50 Hz，而帧频只有25 Hz。因为每一场的行数只是一幅画面行数的1/2，与**逐行扫描**[即50次中的每一次都扫遍所有的行，如图2-9(b)所示]相比，其扫描速度只是后者的一半。由(2-11)式可知，在传送细节相同的条件下，采用隔行扫描时所需要的视频带宽为逐行扫描的1/2。

人们很自然地会联想到，隔多行扫描不是可以进一步降低扫描速度和视频带宽吗？结论是肯定的。但是光栅中每一行的重复频率太低时，会出现行间闪烁的现象。另外，帧频比25 Hz低得太多时还会出现运动的不连续，因此隔多行的扫描方式不为广播电视所采用。



（a）隔行扫描　　（b）逐行扫描

图 2-9　隔行扫描与逐行扫描

2.2.3　电视信号的带宽

在隔行扫描中，场频 $f_v=2f_f$，其中 f_f 为帧频率。从图 2-9 中可以看到，电子束在扫完每一场的最后一行之后，要回到顶端开始下一场的扫描。从下端回到顶端所用的时间，称为**场扫描逆程时间**。设 k_1 为**场扫描正程时间**与完成一场扫描（场正程＋场逆程）的总时间之比，即

$$k_1=\frac{\text{场扫描正程时间}}{\text{场扫描周期}} \tag{2-12}$$

如前所述，电子束在结束行正程扫描以后，也需要有一定的逆程扫描时间才能开始下一行的正程扫描。设 k_2 为行扫描正程时间与扫描一行的总时间（行正程＋行逆程）之比，即

$$k_2=\frac{t_{hf}}{T_h} \tag{2-13}$$

这里，T_h 为行扫描周期。

假设每帧（两场）图像的总扫描行数（包括正程和逆程）为 Z，帧正程扫描（有效）行数为 $n=k_1Z$。我们注意到电子束在进行扫描时，并不是任何情况下每一扫描行都能代表系统在垂直方向上一行的分解力的。例如图 2-10 所示的由黑白相间的水平条纹组成的图像，如果扫描光栅的位置刚好是前一行扫过白条纹，后一行扫过黑条纹，那未扫描的行数等于它在垂直方向上所能分辨的黑白条纹的数目。如果电子束扫描的位置恰好跨在黑白条纹之间，系统则完全失去了分辨这些条纹的能力，这时电视屏幕上显示出来的将是一片灰色。很明显，对于这种极端的情况，只有将黑白条纹的密度降低一半，或者有效行数 n 提高一倍，电视系统才能将其分解清楚。这时系统在垂直方向上的分解力等于有效行数的 1/2。虽然像图 2-10 所示的这种极端的情况在实际景物中并不会经常遇到，但是按统计规律来讲，电视系统在垂直方向上能够分辨的线数（黑白条纹）N_v 要比有效行数 n 低，即

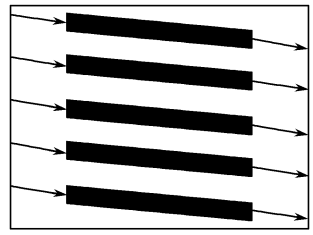


图 2-10　黑白相间的水平条纹与扫描光栅的相对位置

$$N_v=Kn \tag{2-14}$$

式中，K 称为**凯耳**（Kell）**系数**。在实际的系统设计中，K 取 0.7。

在垂直或水平方向上能够分辨的**线数**是电视业界传统用来描述电视系统分辨细节能力的参数，也称垂直或水平分解力。如果我们将分解力换算成空间频率，则(2-14)式变为

$$f_{y\max}=\frac{N_v}{2H}=\frac{Kn}{2H} \tag{2-15}$$

式中，$f_{y\max}$ 为系统可分辨的最高垂直空间频率；H 为屏幕的高度。由于黑白两条线构成一个亮度变化周期，因此在将分解力转换成空间频率时应除以 2。

人眼在视觉敏锐的视场范围内，其垂直和水平方向上的分辨力是近似相同的，因此要求电视系统在这两个方向上也具有相等的分解力，或者说，系统在两个方向上的空间截止频率应该相同。即

要求

$$f_{x\max} = f_{y\max} \tag{2-16}$$

将(2-15)式和(2-16)式代入(2-11)式得到电视信号的上限频率为

$$f_{\max} = \frac{W}{H} \cdot \frac{K}{2} \cdot \frac{n}{t_{hf}} \tag{2-17}$$

式中，W/H 为屏幕的宽高比。根据帧扫描行数、行频和帧频的关系以及(2-13)式，(2-17)式可写成

$$f_{\max} = \frac{W}{H} \cdot \frac{Kk_1}{2k_2} \cdot Z^2 f_f \tag{2-18}$$

上式给出了电视图像信号的上限频率，它在数值上等于视频信号的带宽，因为视频信号的下限频率为零频。在现行电视制度下，由于采用隔行扫描，上式中 f_f 取 25 Hz。假如是逐行扫描，f_f 则需要取 50 Hz 才能使闪烁的感觉减弱到令人满意的程度。使用隔行扫描将视频带宽降低了 1 倍。

总结这一节的讨论，应该牢固地建立起这样的概念：图像的某一细节部分与特定的空间频率相对应，这一信息经过扫描变成电信号(视频信号)，扫描速度越高(帧频越高或每帧画面的扫描行数越多)，这一细节所对应的电信号的频率也越高，传送这一细节所需要的通道带宽也就越高。

2.2.4 彩色空间的处理

从 2.1.4 节中已经知道，三种基色光的强度 R、G、B 之和代表了它们合成彩色的亮度，而它们之间的比值 $R:G:B$ 则代表了合成彩色的色调和饱和度。也就是说，亮度、色调与饱和度相互关联，共同由 R、G、B 三个量来代表。

在彩色电视发展的初期，社会上已经存在着相当数量的黑白电视机和黑白电视台，为了扩大节目的收看率，要求彩色电视系统的设计必须考虑到与已有的黑白电视之间的兼容。所谓**兼容**，即是彩色电视机能收看到黑白电视台播送的节目，而黑白电视机也能收看彩色电视台播送的节目。当然，在这两种情况下所收看到的兼容节目都只能是黑白图像。为了满足兼容的要求，需要将表示亮度和表示颜色的信号分离开，如图 2-11所示，这可以将 R、G、B 通过线性变换来实现。在 PAL 电视制式中，线性变换后产生**亮度信号** Y 和两个**色差信号**，U 和 V。对于黑白图像，$U=0$，$V=0$。对于彩色图像，U 和 V 的比值决定**色调**，而 $\sqrt{U^2+V^2}$ 代表彩色的**饱和度**。对于 NTSC 制式，经类似的变换将产生亮度信号 Y 和色差信号 I、Q[7]。

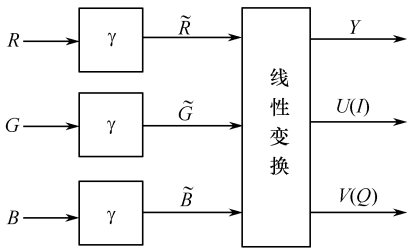


图 2-11 从 R、G、B 到 Y、U、V 的转换

由于摄像管的光电转换特性和显像管的电光转换特性都存在非线性，为了使最终显示出来的光像的亮度层次不出现畸变，必须在将 R、G、B 电信号送上显像管之前进行非线性校正，这称为 γ **校正**。为了降低接收机的成本，γ 校正通常预先在摄像机内进行。图 2-11 中 $\tilde{R}$、$\tilde{G}$、$\tilde{B}$ 信号即为经过 γ 校正的电信号。

图 2-11 中的线性变换实际上是一个解相关的过程。经变换后，表示亮度和表示彩色的量被分离开，这不但有利于彩色与黑白电视的兼容，而且可以利用视觉对彩色的分辨力低于对亮度细节的分辨力的特点，将色差信号用比亮度信号窄的频带传送。例如，在我国的电视制式(PAL)下，Y 的带宽为 6 MHz，U 和 V 的带宽仅为 1.3 MHz。

(2-19)式给出了线性变换的一种具体形式：

$$\begin{pmatrix} Y \\ U \\ V \end{pmatrix} = \begin{pmatrix} 0.299 & 0.587 & 0.114 \\ -0.169 & -0.331 & 0.5 \\ 0.5 & -0.419 & -0.081 \end{pmatrix} \begin{pmatrix} \tilde{R} \\ \tilde{G} \\ \tilde{B} \end{pmatrix} \tag{2-19}$$

式中，$\tilde{R}$、$\tilde{G}$、$\tilde{B}$ 均为值在 0～1 范围内的模拟信号，而 Y 的取值范围为 0～1，U 和 V 的取值范围为 −0.5～0.5。

对于不同的标准白光源和不同色坐标的荧光粉，国际电联无线电组(ITU-R)和国际电视工程师协会(SMPTE)规定了若干种变换矩阵[9]。(2-19)式中的变换矩阵对应于 SMPTE 170 M 三色坐标和参考白色 D_{6500}。

2.2.5 全彩色电视信号

1. 色度信号的传送与彩色电视制式

黑白电视中只需要传送一个亮度信号，而在彩色电视中，则需要在满足与黑白电视兼容，而且在不增加黑白电视所规定的信道带宽(如 6 MHz)的条件下，同时传送亮度信号和两个色差信号。如何找到一个可取的方案去实现这一要求，是彩色电视制式所要解决的问题。不同的色差信号传送方案就形成了不同的彩色电视制式。

当前世界上主要的彩色电视制式有三种：NTSC、PAL 和 SECAM。我国采用的是 PAL 制，而 NTSC 制是 PAL 制的基础。

2. 频谱交错原理和平衡正交调制

众所周知，周期性信号的频谱是线状的，其谱线分布在基频(周期的倒数)和它的谐波上。由于扫描，电视信号在行和场的周期上都呈现着一定的周期性，因此视频信号的能量主要分布在行扫描频率 f_h 及其各次谐波 nf_h 上(见图 2-12)；在两条相邻谱线之间，能量则很微弱，以至于可以把它看作是空白的。色差信号 U 和 V 的频谱分布也遵循同样的规律。如果选择某一数值等于 $(2n+1)f_h/2$(半行频的奇数倍)的载频 f_{sc}，先将两个色差信号 U 和 V 调制到 f_{sc} 上，然后再与亮度信号叠加在一起，色度信号的能量就刚好落在亮度信号频谱的空白处，如图 2-12虚线所示。这就是亮度信号与色度信号按照**频谱交错**间置、共频带传送的基本原理。

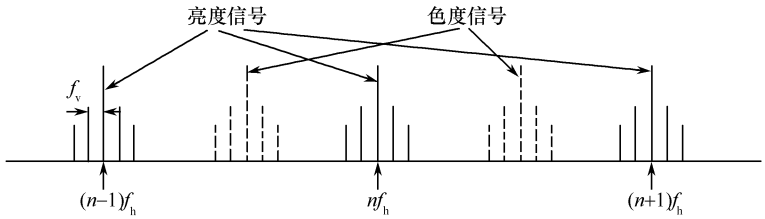


图 2-12 亮度信号与色度信号的频谱交错原理

发射机中用以传送整个彩色电视信号的射频载频通常称为**主载频**，上述用来形成能够与亮度信号频谱间置的色度信号所用的载频 f_{sc} 则称为**副载频**。

因为副载频只有一个，而作为调制信号的色差信号却有两个，这就需要对同一载频的两个不同的相位进行两相调制。在彩色电视中，是将两个色差信号分别调制在副载频的两个正交相位上的，因此称为**正交调制**。

一般调幅波可以表示成 $A(1+m\cos\Omega t)\cos 2\pi ft$ 的形式。将它分解开来则可看出，它包含着载频和上、下边频，载频分量不携带任何信息。为了避免副载频对电视图像的干扰，U 和 V 信号采取了抑制载频分量的调幅方式，称为**平衡调制**。U 和 V 信号经过平衡调制后的结果分别为 $U\sin 2\pi f_{sc}t$ 和 $V\cos 2\pi f_{sc}t$，二者之和称为**色度信号**。

将色度信号与亮度信号 E_y 叠加到一起，就构成 NTSC 制的基带彩色电视信号：

$$E_M = E_y + U\sin 2\pi f_{sc}t + V\cos 2\pi f_{sc}t \tag{2-20}$$

基带彩色电视信号 E_M 的频谱如图 2-13 所示。亮度信号带宽为 0～6 MHz，而色度信号经副

载频 f_{sc} 的调制，在频带高端与亮度信号的频谱交错间置。

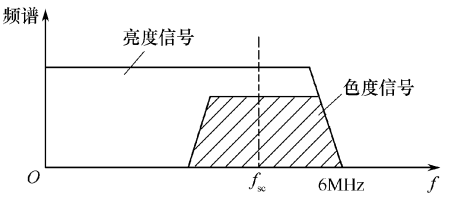


图 2-13　共频带的亮度信号和色度信号

在正统的 NTSC 系统中，是用 I、Q 信号代替上式中的 U 和 V 信号进行正交平衡调制的。I、Q 信号可由 U 和 V 信号通过适当的线性组合而得到[7]，其目的在于利用人眼对彩色平面上不同方向的分辨力不同，进一步降低色差信号所需要的带宽。

NTSC 制的缺点是对信道的**微分相位失真**(DP)比较敏感。传输信道的 DP 失真将导致图像的色调失真，PAL 制就是为了克服这一缺点而提出来的。其具体做法是将(2-20)式中的第 3 项(色度信号的 V 分量)逐行倒相 180°，即

$$E_M = E_y + U\sin 2\pi f_{sc}t + S(t)V\cos 2\pi f_{sc}t \tag{2-21}$$

式中，$S(t)$是一个重复周期等于 2 行扫描时间的方波脉冲，前半个周期(一行的时间)为+1，后半个周期则为−1。对于 PAL 制的详细讨论见文献[7]。

3. 复合电视信号、分离式电视信号和分量电视信号

彩色视频信号再加上有关的(行、场和副载频)同步信号就构成了全彩色电视信号，或称**复合电视信号**(Composite TV Signal)。图 2-14 表示出了对应于彩条图像(白、黄、青、绿、紫、红、蓝、黑)的全彩色电视信号一行的波形，其中左侧的负脉冲和副载波群分别为在行逆程中传送的行同步和色同步信号，正程信号中不同的台阶代表不同彩条的亮度电平，叠加在上面的是经副载频调制后的色度信号。

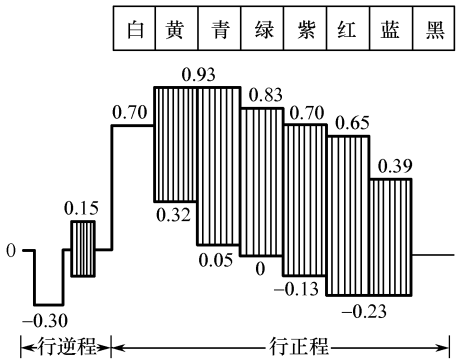


图 2-14　彩条图像的复合电视信号

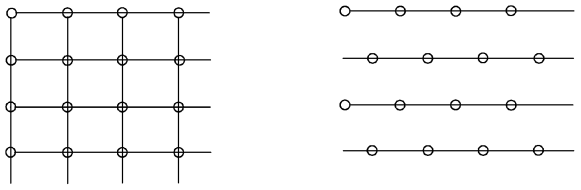
频谱交错是彩色电视发展中的一个卓越思想，但是利用这一原理混合在一起的亮度和色度信号，一般很难在接收端完全地分离。若系统存在非线性，则亮度和色度信号交叉调制将相互形成干扰，使显现出来的图像质量下降。为了避免这一问题，在允许基带传送的情况下，例如录像机或摄像机直接与接收机相连时，常常采用 2 个通道分别传送亮度信号和色度信号，这两个信号就称为**分离式电视信号** S-Video(Seperate Video)。

与复合电视信号和分离式电视信号相对应，未经副载频调制的三个图像信号，R、G、B 或 Y、U、V 则称为**分量电视信号**(Component Signal)。这是在数字电视和多媒体系统中视频信号所采用的基本形式。

2.3　分量彩色电视信号的数字化

取样频率和量化级数是对视频信号进行数字化的两个关键性参数。在数字视频设备中，对亮度信号采用 8 bit 量化(即人们熟知的 256 个灰度级)，是公认比较合理的选择。不过在要求较高的场合，也可以采用 10 bit 或 12 bit 的量化。对色差信号，量化比特数可以取 8 bit，也可以取得低些。

根据取样定理，取样频率应不小于信号最高频率的 2 倍，在实际应用中，一般选取 2.2～2.7 倍。在 PAL 制中，亮度信号和色差信号的最高频率分别为 6 MHz 和 1.3 MHz；在 NTSC 制中，亮度信号的最高频率则为 5.5 MHz。除了满足取样定理的要求外，还希望取样点的位置在每一帧中都是固定的，例如形成图 2-15(a)所示的**正交结构**，或(b)图所示的**梅花点结构**(Quincunx)。取样点在扫描光栅上组成的二维点阵，称为**取样结构**。一般选取固定正交的取样结构，样点在每一行中的位置是相同的(例如，第一个样点总在行正程的起始处)，以便于后续的对数字分量电视信号的处理。



（a）正交结构　　（b）梅花点结构

图 2-15　取样结构示意图

不难理解，要形成正交的取样结构，取样频率应为行频 f_h 的整数倍。在每帧 625 行、每秒 50 场的 PAL 制和每帧 525 行、每秒 60 场的 NTSC 制电视中，625×50 和 525×60 两个乘积几乎相等，二者扫描 1 行所用的时间几乎是一样的，两种行频的最小公倍数是 2.25 MHz。只要将亮度信号和色差信号的取样频率数值取为 2.25 MHz 的整数倍，就可能在两种系统中都得到固定的正交取样结构。

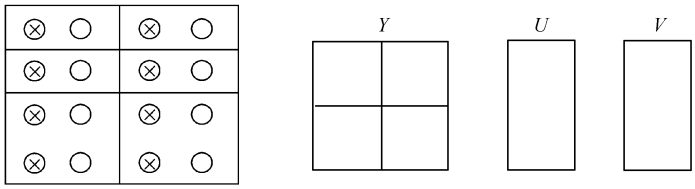
在国际电联无线电组(ITU-R)制定的标准 CCIR 601[10] 中，亮度和色度信号的取样频率，f_y 和 f_c，分别规定为：

$$\begin{aligned} f_y &= 858 f_{\text{h NTSC}} = 864 f_{\text{h PAL}} \approx 13.5\ (\text{MHz}) \\ f_c &= \frac{1}{2} f_y \approx 6.75\ (\text{MHz}) \end{aligned} \tag{2-22}$$

式中，$f_{\text{h NTSC}}$、$f_{\text{h PAL}}$ 分别为 NTSC 制和 PAL 制的行频。将亮度信号和色度信号都量化到8 bits，三个分量信号数字化后所产生的总比特率为：

$$8\ \text{bits} \times (13.5 + 2 \times 6.75) \times 10^6/\text{s} = 216\ (\text{Mb/s}) \tag{2-23}$$

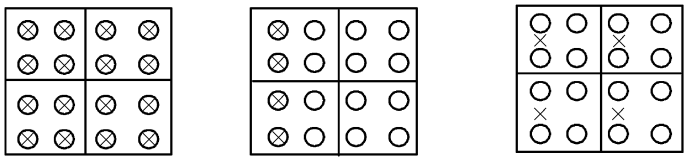
CCIR 601(或 ITU-R BT. 601)还规定 PAL 制每帧的正程有 576 行，NTSC 制则为 480 行，两种制式的行正程都取 720 个样点，这也常称为 **D1 格式**。图 2-16(a)给出了对应的取样结构，其中○点和×点分别代表亮度信号和色差信号取样点的位置；图 2-16(b)给出了取样后亮度和色差信号样点所构成的矩阵的大小。由于两个色差信号的取样频率均为亮度信号取样频率的 1/2，这种格式被称为 4∶2∶2 格式。



（a）4∶2∶2取样结构　　（b）4∶2∶2取样点阵大小

图 2-16　4∶2∶2 格式

根据不同的应用，分量信号的数字化还可以采用 4∶4∶4 和 4∶1∶1 的取样格式，如图 2-17(a)和(b)所示。在这两种格式中，色差信号的取样频率分别与亮度信号取样频率相同，或等于它的 1/4。当对图像质量的要求不太高时(如现场新闻报道)，也可以采用2∶1∶1的取样格式，此时亮度信号的取样频率为 6.75 MHz，色差信号的取样频率为它的 1/2。



（a）4:4:4　　（b）4:1:1　　（c）4:2:0

图 2-17　不同的取样格式

在视频图像的压缩编码中，最常使用的是 4∶2∶0 的取样格式，如图 2-17(c)所示。在这种格式中，亮度和色差信号的取样频率与 4∶2∶2 格式相同，但色差信号的行数为 4∶2∶2 格式的 1/2，因而色差信号样点所构成的矩阵大小只是亮度矩阵的 1/4。

CCIR 601 标准还规定了分量信号的取值范围：

$$Y_d = 219\,Y + 16 \tag{2-24}$$

$$C_b = 224\,U + 128 \tag{2-25}$$

$$C_r = 224\,V + 128 \tag{2-26}$$

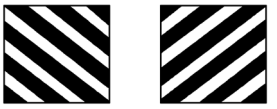
式中，Y 为对取值在 0～1 之间的模拟亮度信号进行数字化之后的结果，U 和 V 为对取值为 −0.5～0.5 之间的模拟色差信号进行数字化之后的结果。经过上式的转换并取最接近的整数值，得到标准的分量信号 Y_d，C_b 和 C_r。Y_d 的取值范围为 16～235，C_b 和 C_r 则为 16～240。

习　题　二

1. 什么叫空间频率？在不涉及和涉及观察者的两种情况下，图像空间频率的高、低各与哪些因素有关？
2. 请设计一个实验，用以测量得到图 2-4 的视觉空间频率响应。
3. 在一个 0.1m×0.1m 的平面内，画出下列函数的亮度分布图，其中 $f_{x0}=25$ 周/m，$f_{y0}=30$ 周/m：

 (1) $L(x)=\frac{1}{2}[1+\cos 2\pi f_{x0}x]$　　(2) $L(x)=\frac{1}{2}\left[1+\cos\left(2\pi f_{x0}x+\frac{\pi}{2}\right)\right]$

 (3) $L(x)=\frac{1}{2}[1+\cos 2\pi f_{y0}y]$　　(4) $L(x)=\frac{1}{2}\left[1+\cos\left(2\pi f_{y0}y-\frac{\pi}{6}\right)\right]$

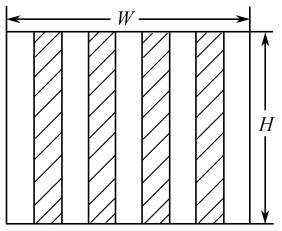


习题 4 图

4. 假设一个 0.1m×0.1m 的屏幕上显示出图示的正弦光栅。
 (1) 分别写出它们包含的频率分量；
 (2) 写出光栅的周期和方位角与空间频率 f_{x0} 和 f_{y0} 之间的一般关系式；
 (3) 负的时间频率有物理意义吗？负的空间频率呢？
5. 已知一个亮度分布函数 $f(x,y)=\{1+\cos[2\pi(f_{x0}x+f_{y0}y)]\}$，其中 $f_{x0}=3$ 周/cm，$f_{y0}=4$ 周/cm，请你
 (1) 在 $x-y$ 平面上画出 1cm×1cm 区域内的二维亮度分布图；
 (2) 求 $f(x,y)$ 的二维空间频率谱 $F(f_x,f_y)$，并在 f_x-f_y 平面上标出各频率分量的位置；
 (3) 用取样频率 $f_{sx}=2$ 周/cm、$f_{sy}=5$ 周/cm 对 $f(x,y)$ 取样，在 f_x-f_y 平面上标出取样后信号频谱 $F'(f_x,f_y)$ 的各频率分量的位置；
 (4) 用下面的低通滤波器对 $F'(f_x,f_y)$ 进行滤波

$$L(f_x,f_y)=\begin{cases}1 & |f_x|\leqslant\frac{f_{sx}}{2},\ |f_y|\leqslant\frac{f_{sy}}{2}\\ 0 & \text{其他}\end{cases}$$

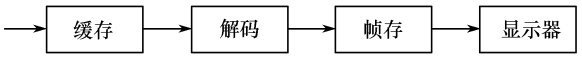
 画出经滤波后的重建亮度分布图。
6. 在我国 625 行/帧、25 帧/秒的隔行扫描的电视制式下，行、场扫描正程时间与行、场总周期的比值约为 $k_2=0.81$ 和 $k_1=0.92$，屏幕宽高比 $\alpha=W/H=4/3$。假设电视机屏幕上显示如右图所示的黑白正弦光栅图案，请你
 (1) 求出对应于该图像的电视信号的最高频率；
 (2) 当人在距离屏幕 6 H 远处观看电视时，如果不断地改变屏幕上栅条的密集程度，当屏幕上有多少条光栅时人眼分辨得最清楚？光栅增加到大约多少条时，人只能看到模糊一片了？
 (提示：根据图 2-4，分别取 4 周/度和 50 周/度为图中曲线的峰值点和截止点对应的横坐标位置。)



习题 6 图

7. 求 1152 行/帧、25 帧/秒的隔行扫描高清晰度电视(HDTV)信号的带宽(假设 k_1，k_2 与第 6 题中相同，$\alpha=16/9$)。如果改为逐行扫描，此时信号的带宽为多少？
8. 电视系统中帧率是根据哪些因素确定的？如果在一个多媒体系统中，网络带宽只容许传输 5 帧/秒的图像信号，在

接收端利用如下图所示的系统(其中显示器为逐行扫描的设备),如何将数据显示才能达到可能达到的最好效果?



习题 8 图

9. 电视系统(或多媒体系统)中为什么要进行 γ 校正? 这项校正通常在系统的哪个部分进行?

10. 通过必要的公式与图表,说明在彩色电视中,如何将带宽都是 0～6 MHz 的 R、G、B 三个单色图像信号,组合成一个能够通过 0～6 MHz 通道传送的彩色电视信号。

11. 在 PAL 彩色电视制式中,行频为 15625 Hz,场频为 50 Hz,副载频 f_{sc} 为 4.43361875 MHz,写出获得正交取样结构的亮度信号的精确取样频率值。

12. 画出在 4∶1∶1 格式中,亮度信号和两个色差信号取样后所形成的矩阵的大小。与 4∶2∶0 格式相比,对于每一帧图像,两种格式所产生的数据量是否相同? 垂直和水平分辨率是否相同?

13. 对带宽为 6MHz 的亮度信号以 13.5MHz 取样,色差信号以 6.75MHz 取样,所得的数字 YUV 信号具有 4∶?∶? 的格式?

参 考 文 献

[1] A. N. Netravali and B. G. Haskell, Digital Pictures: Representation, Compression and Standards, 2nd Ed., Plenum Press, New York, 1995

[2] F. W. Campbell, "The Human Eye as an Optical Filter," Proc. IEEE, Vol. 59, No. 6, 1968, PP. 1009-1014

[3] D. H. Kelly, "Visual Response to Time-Dependent Stimuli", J. Opt. Soc. Amer., 1961, Vol. 51, pp. 422-429

[4] A. J. Seyler and Z. L. Budrikis, "Detail Perception after Scene Changes in a Television Image Presentations," IEEE Trans. IT, Vol. IT-11, No. 1, 1965, pp. 31-43

[5] M. Miyatara, "Analysis of Perception of Motion in Television Signals and its Application to Bandwidth Compression," IEEE Trans. Communications, Vol. Com-23, No. 7, 1975, pp. 761-766

[6] B. Grassman, "On the Theory of Compound Colors", Phil. Mag., Ser. 4, 1854, Vol. 7, PP. 1135-1172

[7] 孙景鳌,蔡安妮. 彩色电视基础. 北京:人民邮电出版社,1996

[8] [美]冈萨雷斯著. 数字图像处理(第二版). 阮秋琦等译. 北京:电子工业出版社,2003

[9] ITU-R Rec. H. 262 ISO/IEC 13818-2, Information Technology——Generic Coding of Moving Pictures and Associated Audio Information, Part 2: Video, 1995

[10] ITU-R Recommendation BT. 601, Encoding Parameters of Digital Television for Studios, Rev. 2, 1990

第 3 章　数据压缩的基本技术

3.1　概述

数据压缩技术的研究已有几十年的历史。从基本原理来看，压缩技术可以分为两大类。第一类方法是基于香农(Shannon)理论[1]的。在这类方法中，视频图像序列利用在空间上和时间上取样得到的一组像素(灰度、彩色)值来表示(声音用在时间上对波形取样的一系列样值表示)；而压缩的方法则是采用一般信号分析的方法来消除数据中的冗余，最终使得用来表示图像的一组数据是互不相关的，且是最简约的。对于这些方法，重要的是了解信源的统计特性，而不关心图像的具体内容，也不考虑或较少考虑人的视觉特性。因此，此类方法称为**基于像素**(或**基于波形**)的压缩方法，也称为第一代图像压缩编码方法。

第一代图像压缩编码方法在 20 世纪 80 年代初已趋于成熟[2]，许多优秀成果已被收入近年来制定的有关图像和视频数据压缩的国际标准，如 JPEG，MPEG 和 H.26X 等。由于希望对图像数据进行几十倍，以至于百倍以上的压缩，采用单一的压缩方法往往不能奏效，因此，各种国际标准都综合利用了多种基本压缩方法来达到所要求的压缩比。当需要进行极低码率的图像数据压缩时，第一代技术往往不能提供令人满意的解码重建图像。

“第二代图像压缩编码方法”这一术语是在 20 世纪 80 年代中期正式出现的[2]。极低码率的图像数据压缩往往采用第二代技术。这类方法在很大程度上依赖于对人类视觉特性的研究，其核心思想是力图发现人眼是根据哪些关键特征来识别图像或图像序列的，然后根据这些特征来构造图像模型。例如，根据人眼对物体的轮廓比对物体内部细节更为敏感的特点，可以利用物体(而不是像素)的集合来表示图像。所谓“物体”是指按边缘信息将某特定图像分割成的若干区域，每个区域内部具有相同的特性(如同一灰度、纹理或运动速度等)。分别对这些区域进行编码将比基于像素的编码方式有效得多。根据视觉特性的其他特点，还可以构造其他的图像模型和编码方式。

鉴于第二代技术尚未达到成熟的阶段，在有关的图像和视频压缩编码的国际标准中也未大量应用，因此，在本章中我们着重讨论第一代技术。

3.2　数据压缩的理论依据

在讨论数据压缩的时候，需要涉及现代科学领域中的一个重要分支——信息论。香农所创立的信息论对数据压缩有着极其重要的指导意义。它一方面给出了数据压缩的理论极限，另一方面又指明了数据压缩的技术途径。本节将对无信息损失条件下数据压缩的理论极限作一简要的介绍；而下一节将讨论限定失真条件下数据压缩的理论极限。

3.2.1　离散信源的信息熵

在日常生活中，当我们收到书信、电话或看到图像时，则说得到了消息，在这些消息中包含着对我们有用的信息。通常，消息由一个有次序的符号(如状态、字母、数字、电平等)序列构成。例如，一封英文信是利用由 26 个英文字母加上标点符号所构成的序列来传递消息的。一个符号所携带的**信息量** I 用它所出现的概率 p 按如下关系定义：

$$I = \log(1/p) = -\log p \tag{3-1}$$

上述定义符合我们日常生活中的概念。例如，如果符号出现的概率为1，这说明传递给我们的是一条几乎肯定要发生的事件的消息，对我们来说是“不出所料”，没有什么信息量；因此，$I=0$。反之，概率较小的符号（事件），出现的不肯定性大，那么收到这个事件发生的消息时，带给我们较大的信息量。

当(3-1)式中的对数以2为底时，它的单位是比特。从后面的讨论中将会看到，表示信息量的比特其含义与二进制符号中的比特并不完全相同。

若信息源所产生的符号取自某一离散集合，则该信源称为**离散信源**。离散信源 X 可以用下式来描述：

$$X = \begin{Bmatrix} s_1 & s_2 & \cdots & s_n \\ p(s_1) & p(s_2) & \cdots & p(s_n) \end{Bmatrix}, \quad \sum_{i=1}^{n} p(s_i) = 1 \tag{3-2}$$

式中，$p(s_i)$ 为符号集中的符号 s_i 发生（或出现）的概率。由于信源产生的符号 s_i 是一个随机变量（在符号产生之前，我们不知道信源 X 将发出符号集中的哪一个符号），而信息量 I 是 s_i[或 $p(s_i)$]的函数，因此 I 也是一个随机变量。对于一个随机变量，研究它的统计特性更有意义。考虑 I 的统计平均值

$$H(X) = \langle I[p(s_i)] \rangle = -\sum_i p(s_i)\log_2[p(s_i)] \quad \text{bit/符号} \tag{3-3}$$

式中，〈 〉表示数学期望。

借用热力学的名词，我们把 H 叫做**熵**。在符号出现之前，熵表示符号集中符号出现的平均不肯定性；在符号出现之后，熵代表接收一个符号所获得的平均信息量。因此，熵是在平均意义上表征信源总体特性的一个物理量。

3.2.2 信源的概率分布与熵的关系

由(3-3)式可以看出，熵的大小与信源的概率模型有着密切的关系。如果符号集中任一符号出现的概率为1，则其他符号出现的概率必然为零，信源的平均信息量（熵）则为零。如果所有符号出现的概率都小于1，熵则为某一正值。这说明，各符号出现的概率分布不同，信源的熵也不同。下面我们来求证，当信源中各事件服从什么样的分布时，熵具有极大值，即求解

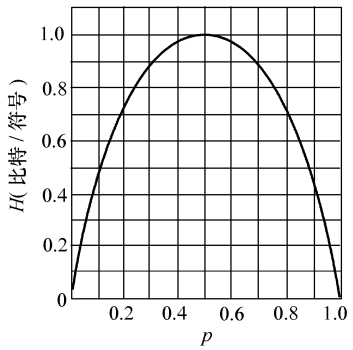


图 3-1 二进制信源的熵与概率 p 之间的关系

$$\text{最大化} \quad H(X) = -\sum_{i=1}^{n} p(s_i)\log_2 p(s_i)$$

$$\text{从属于} \quad \sum_{i=1}^{n} p(s_i) = 1 \tag{3-4}$$

根据求条件极值的拉格朗日乘数法，我们有

$$\frac{\partial\left[H(X) + \lambda\left(\sum_{i=1}^{n} p(s_i) - 1\right)\right]}{\partial p_i} = 0 \quad (i = 1,2,\cdots,n) \tag{3-5}$$

式中，λ 为拉格朗日常数。解方程组(3-5)前，得到

$$p(s_1) = p(s_2) = \cdots = p(s_n) = \frac{1}{n} \tag{3-6}$$

此时，信源具有最大熵

$$H_{\max}(X) = \log_2 n \tag{3-7}$$

这是一个重要结论，有时称为**最大离散熵定理**。

以 $n=2$ 为例，熵随符号“1”的概率 p 的变化曲线如图3-1所示。$p=0$ 或1时，$H(X)=0$。当 $p=1/2$ 时，$H(X)=1$ bit/符号。p 为其他值时，$0 < H(X) < 1$。从物理意义上讲，通常存

储或传输 1 位的二进制数码(1 或 0),其所含的信息量总低于 1bit;只有当字符 0 和 1 出现的概率均为 1/2 时,不肯定性最大,1 位二进数码才含有习惯上所说的 1 比特的信息量。

3.2.3 信源的相关性与序列熵的关系

上面讨论的离散信源所能输出的信息量,是针对一个信源符号而言的。实际上,离散信源输出的不只是一个符号,而是一个随机符号序列(离散型随机过程)。若序列中各符号具有相同的概率分布,该序列(过程)是**平稳的**。若序列中各符号间是统计独立的,即前一个符号的出现不影响以后任何一个符号出现的概率,则该序列是**无记忆的**。

假设离散无记忆信源产生的随机序列包括两个符号 X 和 Y(即序列长度等于 2),且 X 取值于(3-2)式所表示的集合,而 Y 取值于

$$Y=\begin{Bmatrix} t_1 & t_2 & \cdots & t_m \\ q(t_1) & q(t_2) & \cdots & q(t_m) \end{Bmatrix},\quad \sum_{j=1}^{m} q(t_j)=1 \tag{3-8}$$

那么接收到该序列后所获得的平均信息量称为**联合熵**,定义为

$$H(X\cdot Y)=-\sum_i\sum_j r_{ij}\log_2 r_{ij} \tag{3-9}$$

式中,r_{ij} 为符号 s_i 和 t_j 同时发生时的联合概率。由于 X 和 Y 相互独立,$r_{ij}=p(s_i)q(t_j)$,(3-9)式变为

$$H(X\cdot Y)=H(X)+H(Y) \tag{3-10}$$

将上面的结果推广到多个符号的情况,可以得到如下结论:离散无记忆信源所产生的符号序列的熵等于各符号熵之和。要知道收到其中一个符号所得到的平均信息量(即序列的平均符号熵)可以用序列熵除以序列的长度求得。显然,当序列是平稳的,任一符号的熵就是序列的平均符号熵。

假设离散信源是有记忆的,而且为了简单起见,只考虑相邻两个符号(X 和 Y)相关的情况。由于其相关性,联合概率 $r_{ij}=p(s_i)P_{ji}=q(t_j)P_{ij}$,其中 $P_{ji}=P(t_j/s_i)$ 和 $P_{ij}=P(s_i/t_j)$ 为条件概率。

在给定 X 的条件下,Y 所具有的熵称为**条件熵**,即

$$H(Y/X)=\langle -\log_2 P_{ji}\rangle=-\sum_{i=1}^{n}\sum_{j=1}^{m} r_{ij}\log_2(r_{ij}/p(s_i)) \tag{3-11}$$

上式中在对 $-\log_2 P_{ji}$ 进行统计平均时,由于要对 s_i 和 t_j 进行两次平均,所以用的是联合概率 r_{ij}。利用(3-9)式和(3-11)式以及联合概率与条件概率之间的关系,不难证明联合熵与条件熵之间存在下述关系:

$$H(X\cdot Y)=H(X)+H(Y/X)=H(Y)+H(X/Y) \tag{3-12}$$

上式表明,如果 X 和 Y 之间存在着一定的关联,那么当 X 发生,在解除 X 的不肯定性的同时,也解除了一部分 Y 的不肯定性。但此时 Y 还残剩有部分的不肯定性,这就是(3-12)式中 $H(Y/X)$ 的含义。我们把无条件熵和条件熵之差定义为**互信息**,即

$$I(X;Y)=H(Y)-H(Y/X) \tag{3-13}$$

$$I(Y;X)=H(X)-H(X/Y) \tag{3-14}$$

显然,$I(X;Y)=I(Y;X)\geqslant 0$。

两个事件的相关性越小,互信息越小,残剩的不肯定性便越大。当两事件相互独立时,X 的出现,丝毫不能解除 Y 的不肯定性。在这种情况下,联合熵变为两个独立熵之和(见(3-10)式),从而达到它的最大值。图 3-2 给出了无条件熵、条件熵和互信息之间关系的示意。

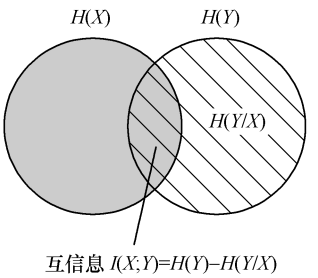


图 3-2 无条件熵、条件熵和互信息之间的关系

由(3-12)式和(3-10)式,可以得到

$$H(X\cdot Y)=H(X)+H(Y/X)\leqslant H(X)+H(Y) \tag{3-15}$$

对于信源输出序列中有多个符号相关的情况，也可以得到类似的结果。序列熵与其可能达到的最大值之间的差值反映了该信源所含有的冗余度。信源的冗余度越小，即每个符号所独立携带的信息量越大，那么传送相同的信息量所需要的序列长度越短，或符号数越少。因此数据压缩的一个基本途径是去除信源产生的符号之间的相关性，尽可能地使序列成为无记忆的。而对于无记忆信源而言，如 3.2.2 节所述，在等概率情况下，离散平稳无记忆信源单个符号的熵（平均信息量）具有极大值。因此，在无信息损失的情况下，数据压缩的另一基本途径是改变离散无记忆信源的概率分布，使其尽可能地达到等概率分布的目的。

3.3 信息率—失真理论

本节将从信息论的角度，对在有信息损失情况下数据压缩的理论极限做较为严谨的讨论。有信息损失情况下的数据压缩，就是将信源输出的数据转化为简化的或压缩的版本，而它的逼真度的损失不超出某一容限。它属于信源编码的范畴。

3.3.1 通信系统的一般模型

图 3-3 为常见的简单通信系统模型。我们将信道编、译码器归入信道，所以信道可以看成是无噪声的。

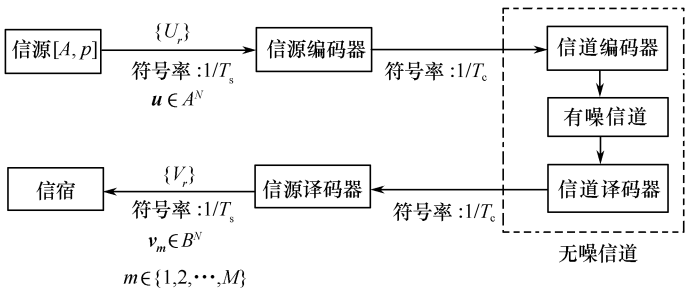


图 3-3 通信系统模型

设信源产生离散随机过程$\{U_r\}$，$r=-\infty,\cdots,-2,-1,0,1,2,\cdots,\infty$，即随机变量序列。$u$ 是信源符号 U 的样，$u\in A$，A 是信源符号集。信源的特征可以用符号集 A 以及相应的概率分布表示。A 可以是实数集，有限集，波形集，二维空间的亮度分布集，等等。概率特征可记为 p，它是一族 N 维概率密度或分布函数。对信源还可能有平稳、遍历、独立、限功率等制约。

设信源为恒速率信源，它每 T_s 秒产生 A 的一个符号，即信源符号率 R_s 为 $1/T_s$。信源编码器将每个长度为 N 的信源符号组编为 K 个编码符号组成的码字，编码符号取自于符号集 E，我们将该码字姑且称为**编码码字**。若 E 中包含 a 个符号，则可能构成的编码码字数 $M=a^K$。设信道为恒速率数字信道，它的传输率为 $R_c=1/T_c$，即每 T_c 秒传输一个编码符号。信源译码器将信道输出还原为长 N 的、其形式可以为信宿接受的符号组，该符号组称为**还原码字**。由于编码码字数为 M，还原码字数也为 M。还原码字串成随机过程$\{V_r\}$，v 是 V 的样，$v\in B$。B 可以等于 A，也可以不同于 A。为了使全系统有协同的符号流量，须使 $NT_s=KT_c\triangleq T$，即在信源产生 N 个符号的期间，信道传输 K 个符号，同时译码器输出 N 个符号。这将保证$\{V_r\}$与$\{U_r\}$有相同的符号率，只不过通常二者之间存在迟延而已。

如果编码器是数据压缩设备，则定义压缩率（或称码的符号压缩率）$\xi=K/N=T_s/T_c=R_c/R_s$，即每个信源符号平均所需的编码符号数。

前面已说明，信道可以是无噪声的，因此可以认为信源序列$\{U_r\}$直接由编、译码器映射为信宿

序列$\{V_r\}$。假设 A 中有 l 个符号，则可能出现的信源符号组 $\boldsymbol{u}$ 有 l^N 种。如前所述，不论 B 是否与 A 相等，构成信宿序列的还原码字 $\boldsymbol{v}_m$ 只有 M 种。为了达到压缩的目的，需要 $M<l^N$。显然，用 M 种码字代表 l^N 种信源符号组(二者长度均为 N)有可能损失信息。

从信息论的观点，逼真度的损失可以用**失真**来度量。定量地表征失真的是失真函数 $d(u,v)$，它是一个二元函数，它确定在信宿用 v 代替信源输出 u 时失真的大小。所以，$d(u,v)$ 常为非负的实数集内的一个数值。由于 U 和 V 都是随机量，$d(u,v)$ 也是随机量，因此通常须定义**平均失真** $\bar{d}$：

$$\bar{d}=\langle d(U,V)\rangle \tag{3-16}$$

这里〈 〉是在 $U\in A$ 和 $V\in B$ 上的数学期望。再考虑到信源输出和信宿输入均为随机序列，可定义长度为 N 的符号组的平均失真为

$$\bar{d}_N=\frac{1}{N}\sum_{r=1}^{N}\langle d_r(U_r,V_r)\rangle,\qquad 1\leqslant N<\infty \tag{3-17}$$

对于不同的 r，$d_r(\cdot)$可以相同，也可以不同。我们将这种以**单符号失真**来衡量逼真度的准则记为 F_d。

数据压缩的最基本问题是，在给定的一类编码器 G_E 和一类译码器 G_D 下，可能达到的最佳性能，或最小平均失真有多大。最小平均失真由下式计算：

$$D^{(N)}(G_E,G_D)=\inf_{(G_E,G_D)}\langle\bar{d}_N\rangle \tag{3-18}$$

其中下确界 inf 是在给定类的所有编、译码器上确定的。因为编、译码器的构造受到实际的限制，所以没有采用极小值，而取下确界。所谓某一类编、译码器是以它们的复杂性或压缩要求，或其他约束条件来划分的，例如，可以指所有的 M 级的量化器、所有的长度一定和符号集大小一定的分组码，或给定形式且输出熵受约束的所有的时不变滤波器等。

上述基本问题也可以逆向提出，即在给定一定的逼真度，或平均失真不超出 D 的条件下，所需传输的最低信息率(按每信源符号计)有多大，从而再据此设计编、译码器。信息论的有关基本定理是从这个角度出发的，因为这样的命题更结合实际，也更严谨。

3.3.2 信息率—失真函数

信息率—失真理论，简称率失真理论，是信息论的一个分支，它研究信源的熵超过信道容量时出现的问题。香农在 1948 年的《通信的数学理论》中开始涉及这一问题，而 Berger 1971 年所著的《信息率失真理论》[3]给读者提供了这方面的系统知识。

根据 3.3.1 节的讨论，设信源输出被分割成长度为 N 的符号组。我们希望每一个符号组(或向量)$\boldsymbol{u}\in A^N$ 通过编、译码被变换为一个在 $\bar{d}_N$ 意义下最佳的 $\boldsymbol{v}\in B^N$(即使 $\bar{d}_N$ 为最小)，其中 $\boldsymbol{u}$ 表示 $(u_1,\cdots,u_N)$，$\boldsymbol{v}$ 表示$(v_1,\cdots,v_N)$。

对于给定的信源，$p(\boldsymbol{u})$，$\boldsymbol{u}\in A^N$ 是确定的。对于每个指定的条件概率分布 $P(\boldsymbol{v}\mid\boldsymbol{u})$，$\boldsymbol{u}\in A^N$，$\boldsymbol{v}\in B^N$，可以得到信宿的概率分布为

$$q(\boldsymbol{v})=\sum_{u\in A}p(\boldsymbol{u})P(\boldsymbol{v}/\boldsymbol{u}),\qquad v\in B \tag{3-19}$$

而 U^N 和 V^N 间的互信息为

$$I(U^N;V^N)=H(V^N)-H(V^N/U^N)=\sum_{\boldsymbol{u}\in A^N}p(\boldsymbol{u})\sum_{\boldsymbol{v}\in B^N}P(\boldsymbol{v}/\boldsymbol{u})\log_2\frac{P(\boldsymbol{v}/\boldsymbol{u})}{q(\boldsymbol{v})} \tag{3-20}$$

因为 $p(\boldsymbol{u})$ 是给定的，互信息量将随 $P(\boldsymbol{v}/\boldsymbol{u})$ 而改变，而 $p(\boldsymbol{v}/\boldsymbol{u})$ 反映了编、解码器的性能。现在可以提出这样一个问题，如果将平均失真 $\bar{d}_N$ 限制在一个规定值 D 以下，$I(U^N;V^N)$ 至少要多大?也就是说，还原码字中至少要包含信源符号组的多少信息才能满足 $\bar{d}_N\leqslant D$?为了回答这一问题，需要定义

信源$[A,p]$相对于F_d的**率失真函数**$R(D)$，即

$$R(D)=\frac{1}{N}\min_{P(\boldsymbol{v}/\boldsymbol{u})\in P_D} I(U^N;V^N) \tag{3-21}$$

其中P_D是满足$\bar{d}_N\leqslant D$的所有条件概率分布$P(\boldsymbol{v}/\boldsymbol{u})$的集合，即

$$P_D=\{P(\boldsymbol{v}/\boldsymbol{u})\mid \bar{d}_N=\sum_{\boldsymbol{v}\in B^N}\sum_{\boldsymbol{u}\in A^N} p(\boldsymbol{u})P(\boldsymbol{v}/\boldsymbol{u})d_N(\boldsymbol{u},\boldsymbol{v})\leqslant D\} \tag{3-22}$$

当信源为平稳、无记忆时，(3-21)式可以简化为

$$R(D)=\min_{P(\boldsymbol{v}/\boldsymbol{u})\in P_D} I(U;V) \tag{3-23}$$

其中

$$I(U;V)=\sum_{u\in A}p(u)\sum_{v\in B}P(v/u)\log_2\frac{P(v/u)}{q(v)} \tag{3-24a}$$

$$P_D=\{P(v/u)\mid \bar{d}_N=\sum_{u\in A}\sum_{v\in B}p(u)P(v/u)d(u,v)\leqslant D\} \tag{3-24b}$$

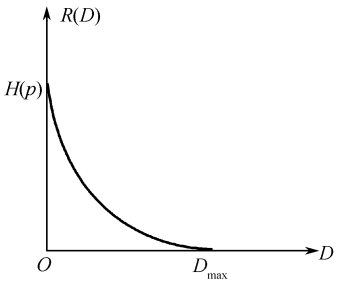


图 3-4　离散信源的率失真函数

这就是说，为了使失真度不大于D，信源序列传送给信宿的最小平均信息率是$R(D)$。这个函数既依赖于信源的统计特性，也依赖于失真的度量，同时与编、解码器的类型$p(v/u)$有关。$R(D)$是在失真不大于D的情况下，对信源信息率压缩的理论极限。

可以证明[3]，尽管在(3-21)式用min，而不用inf，$R(D)$是存在的；同时，$R(D)$在定义域内是单调递减的下凸函数。关于$R(D)$的其他性质，可参阅文献[3,4]。对于离散信源，$R(D)$一般如图 3-4 示，其中$H(p)$为熵函数。

3.3.3　限失真信源编码定理

在实际的通信系统中，用来传送信源信息的信息率远大于$R(D)$函数所规定的值。那么，这个极限值是否能够达到或接近呢？下面介绍的编码定理将回答这个问题。

用G表示解码还原的字集$G=\{\boldsymbol{v}_1,\boldsymbol{v},\cdots,\boldsymbol{v}_M\}$，称$G$为长度$N$、字数为$M$的还原码，$G$的元素称为(还原)码字。现在用$G$对信源$[A,p]$的输出进行编、译码，即每个长度为$N$的信源符号组被映射为某一码字$\boldsymbol{v}\in G$，以使$d_N(\boldsymbol{u},\boldsymbol{v})$为最小，记此最小值为

$$d_N(\boldsymbol{u}\mid G)=\min_{\boldsymbol{v}\in G}\{d_N(\boldsymbol{u},\boldsymbol{v})\} \tag{3-25}$$

G的平均失真为

$$d(G)=\langle d_N(U\mid G)\rangle_p=\sum_{\boldsymbol{u}\in A^N}p(\boldsymbol{u})d_N(\boldsymbol{u}\mid G) \tag{3-26}$$

式中，$\langle\ \rangle_p$表示在$p(\boldsymbol{u})$上的平均。因此，$d(G)$是随机产生的信源符号组$\boldsymbol{u}=(u_1,\cdots,u_N)$以$G$中最近码字为近似时所产生的失真的统计平均值。

字数为M、长为N的还原码G的信息率通常定义为

$$R=\frac{1}{N}\log_2 M\quad(\text{bit/ 信源符号}) \tag{3-27}$$

这是显而易见的，因为还原码字$\boldsymbol{v}$只有M种可能值，其最大可能信息率为$\log_2 M$(bit/码字)。如果$d(G)\leqslant D$，称G是***D* 容许的**。将字数最少的***D* 容许码**记为$M(N,D)$。

有了这些预备知识，我们可以给出如下的信源编码定理和它的逆定理，定理的证明可参考文献[3]。

正定理　对于给定平稳离散无记忆信源$[A,p]$和单符号逼真度准则F_d，设相对于F_d的$[A,p]$的率失真函数为$R(\cdot)$。那么，给定任意$\varepsilon>0$和$D\geqslant 0$，可以找到正整数N，以使一个长度为N的$(D+\varepsilon)$容许码存在，且其信息率$R\leqslant R(D)+\varepsilon$。也就是说，在$N$充分大时，以下不等式成立：

$$\frac{1}{N}\log_2 M(N,D+\varepsilon)<R(D)+\varepsilon \tag{3-28}$$

换句话说，失真接近于 D、而信息率任意接近于 $R(D)$ 的码是存在的。

逆定理 不存在信息率小于 $R(D)$ 的任何 D^- 容许码，即对于所有的 N 下式成立：

$$\frac{1}{N}\log_2 M(N,D) \geqslant R(D) \tag{3-29}$$

简单地说，当失真为 D 时，信息率不能小于 $R(D)$ 。

现在考虑信源编码正、逆定理如何用于图 3-3 所示的系统。假设对这个系统的要求是信宿相对于 F_d 以失真 $D+\varepsilon$ 还原平稳离散无记忆信源$[A,p]$。编码器可以用$(D+\varepsilon)$容许码，将每个长度为 N 的信源输出序列 $\boldsymbol{u}$ 映射为一个码字 $\boldsymbol{v}$，以使 $d_N(\boldsymbol{u},\boldsymbol{v})$ 为最小。因为正定理保证 $R<R(D)+\varepsilon$，所以只要信道容量(以每信源符号计)满足

$$C > R(D)+\varepsilon \tag{3-30}$$

编码器的输出就可以在译码器中以任意小的差错概率还原。图 3-3 中信道(可能包含信道编、译码器)输入、输出的符号则是无关紧要的，因为经过传输所增加的失真度为任意小，所以全系统仍以失真度 $D+\varepsilon$ 工作。另外，每个信源序列 $\boldsymbol{u}$ 映射为码字 $\boldsymbol{v}$ 是确定性的，所以信源译码器放在信道输出对上述结果并不产生影响。由此，可得下面的信息传输定理[3]：

正定理 对于 $\varepsilon>0$，前述离散无记忆信源可以在容量 $C>R(D)+\varepsilon$ 的任何离散无记忆信道的输出端还原，而失真度为 $D+\varepsilon$。

逆定理 前述离散无记忆信源不可能在信道容量 $C<R(D)$ 的任何离散无记忆信道的输出以失真 D 还原。

这两个定理表明，信源和信宿间在限失真 D 下的通信，要求信道容量为 $R(D)$ 既是必要的，也是充分的。若一个系统在信道容量 $C=R(D)$ 时可使平均失真等于 D，则这个系统是**理想的**。在理想系统中，信源编码和信道编码可以完全分开处理。信源编码在给定的信源和逼真度准则下谋求最佳码，而不管信道结构，只要容量相等，任何信道都可以应用。信源编码器的输出的熵在码字充分长时趋近于信道容量。根据信源的渐近等分性(AEP)，在信源序列长度无限增大时，所有典型序列的概率和趋近于 1，且每个序列的概率接近于相等，所以编码器输出的码字也是接近于等概率的(可以舍弃那些总概率接近于零的非典型序列)。当然，这里已经应用了遍历性，通常我们假设所讨论的信源是遍历的。这样，就允许信道编码器针对这些典型序列工作，而不管信源的细节；它只需建立信源码字和信道码字之间的一一对应关系。

以上的讨论只涉及平稳离散无记忆信源。对于其他信源的率失真理论，有兴趣的读者请参阅文献[4]。

3.4 取样频率的转换

我们知道，对于某一给定的模拟信号，取样频率越高，取样后所获得的样值越多，编码后的比特率也就越高。在某些情况下，在已经获得了数字信号之后，可能希望通过降低原有数字信号的取样率来降低信号的码率，例如由于传输带宽的变化，希望从传送高分辨率(720×576)的图像改为传送低分辨率(360×288)的图像。这种由高取样率的样值去推算低取样率样点数值的技术，在数字信号处理中称为**抽取**，或**下取样**。相对应地，由低取样率的样值去估计较高频率的取样点上的数值，这种技术称为**内插**，或**上取样**。如何在降低取样率时尽可能地减少混叠失真，以及如何在内插时获得尽可能准确的估值，是本节讨论的主要内容。

3.4.1 下取样

下取样也称为**亚取样**。图 3-5(a)是一个离散时间(或空间)信号的频谱，其取样频率 f_1 略高

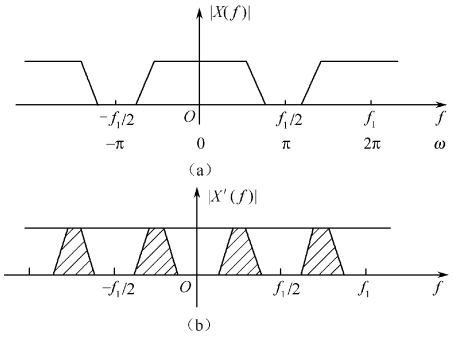


图 3-5　离散时间(或空间)信号的频谱

于奈氏频率。如果我们希望将取样频率降低到 $f_2=f_1/n$(n 为正整数),而 f_2 低于奈氏频率时,不能简单地采取隔 n 个样点抽取一个、丢弃其余样点的做法,因为这样作完全等效于用 f_2 作为取样频率直接对原模拟信号进行抽样,将会造成频谱的混叠。对频谱如图(a)所示的离散时间信号进行 2 抽 1(每隔 1 个样点丢弃 1 个样点)以后造成混叠失真的情况如图(b)所示,图中阴影部分为频谱混叠的区域。当图中的横坐标以对取样频率归一化的角频率 ω 度量时,$0\sim2\pi$ 表示了频谱的一个周期,$[0,\pi)$区间对应于正频率部分,$[\pi,2\pi)$区间对应于负频率部分。在数字信号处理中,对于周期性延拓的频谱,通常只针对 1 个周期进行讨论。

在上面的例子中,一种避免混叠失真的做法是,先用一个截止频率为 $f_1/2$ 的低通滤波器将图 3-5(a)所示频谱的基带部分取出,恢复出原来的模拟信号;然后用另一低通滤波器将此模拟信号的带宽限制到 $f_2/2$,再重新取样得到取样频率为 f_2 的样值[见图 3-6(a)]。实际上,我们可以省略恢复模拟信号的中间步骤,将图中的两个滤波器合并为一个,其截止频率由带宽较窄的滤波器确定($f_2/2$)。此滤波器称为**抗混叠滤波器**。

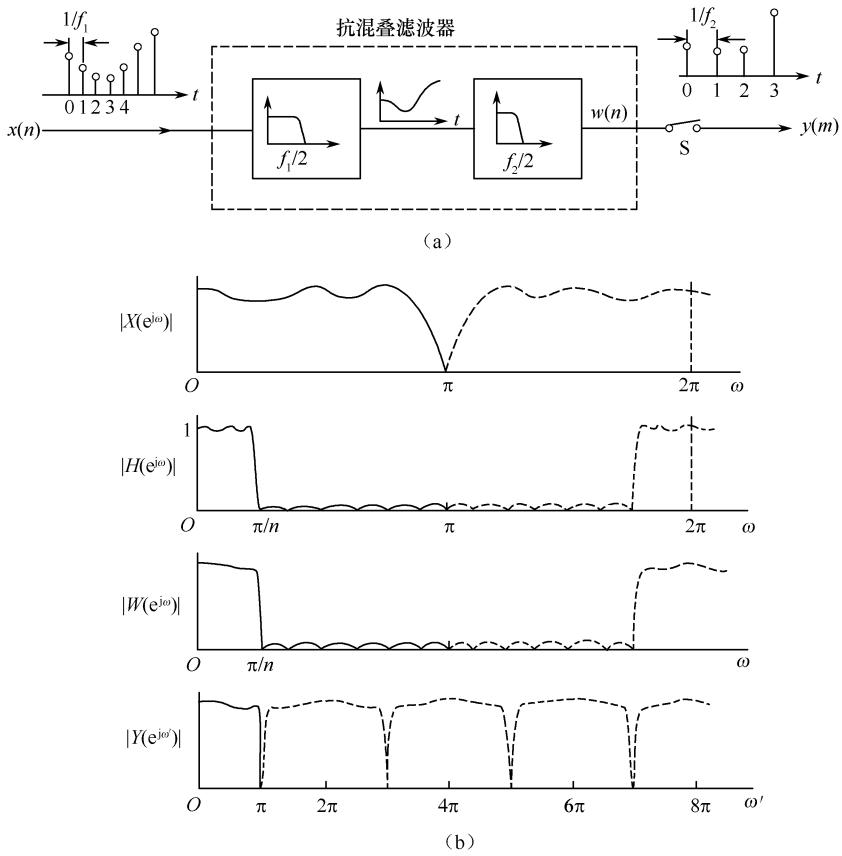


图 3-6　取样频率的降低

图 3-6(b)从频谱的角度描述了下取样的实现过程。其中 $X(e^{jw})$ 是输入信号 $x(n)$的频谱；$|H(e^{jw})|$是抗混叠滤波器的频率响应，它的截止频率为 $f_1/2n$，n 为下取样的比值(图中 $n=4$)；$|W(e^{jw})|$为通过抗混叠滤波器后信号 $w(n)$的频谱，它的最高频率被限制到下取样频率 (f_1/n) 的 1/2；$|Y(e^{jw'})|$为经下取样后的信号 $y(m)$的频谱。值得注意，图中的 ω 和 ω' 是分别相对于 f_1 和 f_2 归一化的。

抗混叠滤波器通常用 FIR 数字滤波器来实现，它的基本构成如图 3-7 所示，其中 z^{-1}代表一个时钟周期的延时。由图可知，滤波器的输出信号为

$$w(n)=\sum_{i=0}^{M-1}a(i)x(n-i) \qquad (3\text{-}31)$$

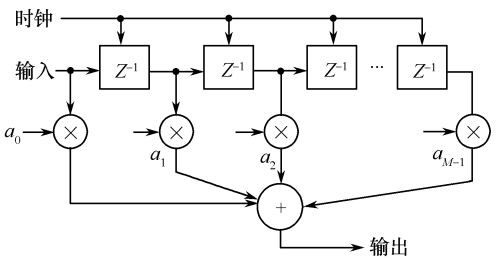


图 3-7　FIR 数字滤波器

不难看出，该滤波器的冲激响应为$\{a(i),i=0,1,\cdots,M-1\}$，是有限长度的。FIR 滤波器在工作时通常要对 $w(n)$每一个样值进行计算(加权、求和)，但在下取样中使用时，有数量众多的样值最终要被丢弃，对这些将要被丢弃的样点值进行计算是一种资源的浪费。如果将 FIR 和再抽样结合在一起，则可以有效地解决这个问题。此时，图 3-7 中的延时单元按输入信号的时钟频率 f_1 工作，而乘法器和加法器则工作在较低的输出信号的时钟频率 f_1/n 上。

3.4.2　上取样

取样定理告诉我们，以不低于奈氏频率取样得到的离散时间(或空间)信号，经过截止频率为 1/2 取样频率的理想低通滤波器，可以完全(无差错)地恢复出原来的模拟信号来。图 3-8 说明了这个过程。由于理想低通的冲激响应为 sinc 函数，信号的每个样值(图中的黑点)通过滤波器后形成一个 sinc 脉冲，这些 sinc 脉冲相互叠加的结果就得到原来的模拟信号 $f(t)$。由此可见，通过理想低通滤波器，我们可以获得输入样值之间任意点的内插值，截止频率为 1/2 取样频率的理想低通滤波器可以称为理想**内插滤波器**。由于理想低通是不能实现的，只能采用矩形窗或其他形式的低通窗截断后的 sinc 函数作为内插滤波器的冲激响应；而且在上取样中，我们只需要得到在上取样频率样点(而不是任意点)上的内插值，因此内插滤波器可以采用 FIR 数字滤波器，并不必须是模拟滤波器。

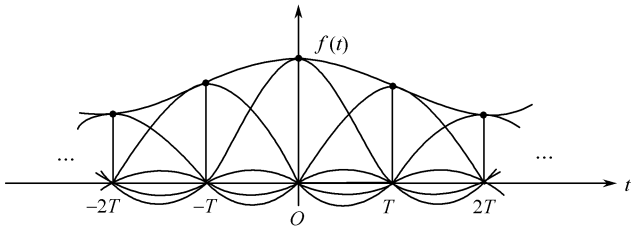


图 3-8　理想内插波形

图 3-9 给出了上取样的实现过程。首先在取样率为 f 的输入序列中插入零值样点，使其达到要求的输出取样率 $f'=Lf$；然后对这些样点上的值进行估计，这可以让此序列通过内插滤波器来实现。图下方左列给出了系统各点的时间波形($L=3$)，右列给出了它们对应的频谱。由图看出，插入零值样点并不改变信号的频谱，而只改变了观察的频率区间。由于取样频率上升了 3 倍，一个用上采样频率 f' 归一化的频谱周期对应于原取样频率 f 下的三个周期。经过内插低通滤波器 $h(m)$之后，只有 $w(m)$的低端频谱保留，从而得到在上取样频率下的信号 $y(m)$。

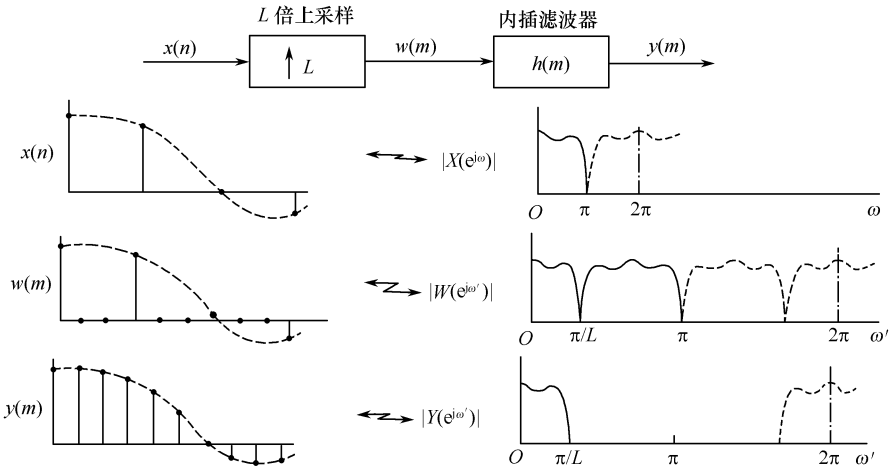


图 3-9　取样频率的增高

假设图 3-9 的内插滤波器的冲激响应 $h(m)$ 如图 3-10(a)所示，这里 m 代表与取样点相对应的时间 t 的离散值。$h(m)$ 的包络是一个 sinc 函数，由于滤波器的截止频率为原取样频率的 1/2，即 $f/2$，所以 $h(m)$ 的主瓣宽度由输入信号的取样周期($1/f$)决定。而滤波器需要内插出取样频率为 $3f$ 的样点值，因此 $h(m)$ 的取样周期应为 $1/3f$。按照数字滤波器的一般设计方法，滤波器的所有单元都应工作在较高的频率 $f'=3f$ 上，但是如果采用图 3-10(b)所示的分相位内插的方法，各单元的工作频率则可以降低到 f。由图看出在这种方法中，滤波器冲激响应 $h(m)$ 由 $h_0(m)$、$h_1(m)$ 和 $h_2(m)$ 三个并行分支组成，$h_1(m)$ 和 $h_2(m)$ 相对于 $h_0(m)$ 分别有 $1/(3f)$ 和 $2/(3f)$ 的延时。取样开关每隔 $1/f'$ 切换一个分支，依次选择不同分支的输出。

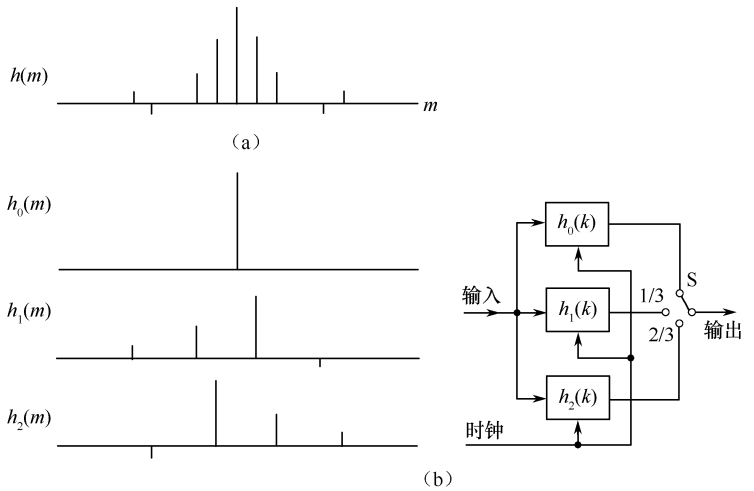


图 3-10　分相位内插滤波器

线性内插是一种广泛使用的内插方法，它利用两相邻样值的加权和内插出二者之间任意位置上的数值，每个样值的权值与该样值到内插位置的距离成反比。假设对图 3-11(a)所示的图像进行 2∶1 上取样，其中，深色方块为输入图像的像素，浅色方块为需要内插出的像素。对于像素 b 有

$$b = \text{round}[(G+H)/2] \tag{3-32}$$

由于 b 与 G 和 H 的距离相等，因此使用了相同的权值。从内插滤波器的角度来看，(3-32)式代表的是一个双抽头的 FIR 滤波器，其冲激响应如图(b)的上一行所示。让我们再来看另一种内插方法：每个半像素位置(如 b,h,m,s)上的值都由一个冲激响应(权值)为{1/32，−5/32，5/8，5/8，−5/

32,1/32}的 6 抽头 FIR 滤波器内插出来。例如，

$$b = \text{round}[(E - 5F + 20G + 20H - 5I + J)/32] \tag{3-33}$$

图(b)的下一行画出了该滤波器的冲激响应。比较图 3-11(b)所示的两个冲激响应，h_2 比 h_1 更接近于理想内插滤波器的冲激响应(sinc 函数)，因此，其对应的内插结果更接近于原模拟信号在新抽样频率上的样值。

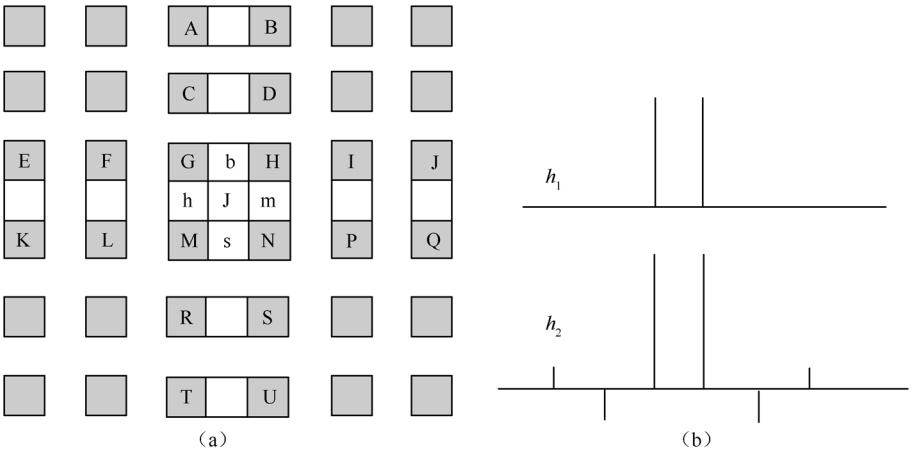


图 3-11　两种 2∶1 内插滤波器的比较

3.4.3　分数比率转换

在有些情况下，上取样或下取样的倍数不一定是整数。此时，取样频率从 $f_1 = mf$ 到 $f_2 = nf$ (m 与 n 之比不为整数)的转换，可分成两个整数比率的转换进行。第一步将 f_1 内插到数值为 f_1 和 f_2 的最小公倍数的频率 mnf 上；第二步再对所得到的序列用 f_2 重新取样。图 3-12表示出这一过程。图中第一个滤波器为内插滤波器，第二个为重新取样前的反混叠滤波器。在实际实现时，这两个滤波器可以合并为一个，其截止率为 f_1 和 f_2 之中较低的一个。

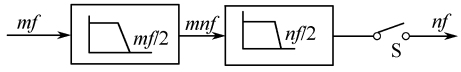


图 3-12　取样频率为分数比率的转换

3.5　预测编码

预测编码根据 3.2 节中介绍的原理，旨在去除相邻像素之间的冗余度，只对不能预测的信息进行编码。这里说的“相邻”可以指像素与它在同一帧图像内上、下、左、右的像素之间的空间相邻关系，也可以指该像素与相邻的前帧、后帧图像中对应于同一空间位置上的像素之间时间上的相邻关系。其中后者，即相邻帧图像中位于同一坐标位置上的像素，常称为该像素的**同位(co-located)像素**。人们在几十年之前就认识到，电视图像的相邻帧之间画面内容变化甚小，存在着大量的重复信息；另外，图像亮度、色度的空间分布多是渐变的，同一帧画面内相邻像素之间也存在着重复信息。但是，要降低像素间信息的冗余度，以降低对传输信道容量的要求，则只有在数字电视中才有了实际实现的可能。

3.5.1　差分脉冲编码调制(DPCM)

差分脉冲编码调制简称为 **DPCM**(Differential PCM)，它是预测编码的一种基本方式。图 3-13 为 DPCM 的原理方框图。

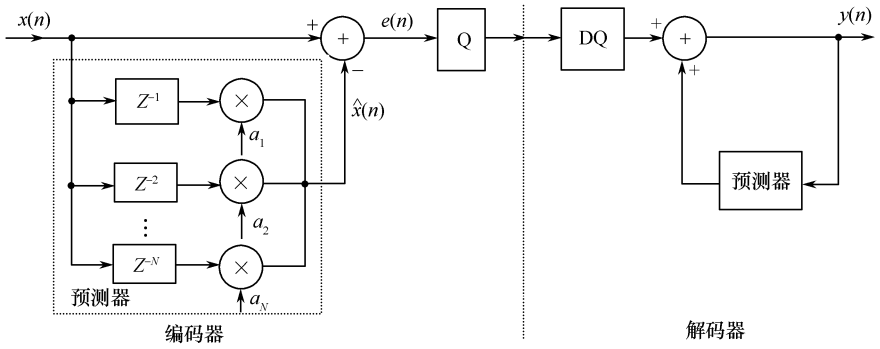


图 3-13　DPCM 的原理方框图

输入信号 $x(n)$ 是取样后(没有量化以前)的图像信号的样值,虚线方框内的电路称为**预测器**,其中 $z^{-i}(i=1,\cdots,N)$ 为延时单元,其延迟时间为 i 个取样周期,$a_1,a_2,\cdots,a_N$ 为固定的加权系数,其数值在预测器设计中确定。预测器根据前几个邻近样值(数目为 N)推算出当前样值的估计值 $\hat{x}(n)$。Q 为量化器。发送端只对预测的误差信号[即当前的实际样值 $x(n)$ 与估计值 $\hat{x}(n)$ 之差]$e(n)$ 进行量化编码、传送,而不是传送 $x(n)$ 本身。

图 3-13 中预测器的输出为

$$\hat{x}(n)=\sum_{i=1}^{N}a_i x(n-i) \tag{3-34}$$

式中,N 称为预测器的**阶数**。由于 $\hat{x}(n)$ 等于各输入样值的线性叠加,该预测器称为**线性预测器**。

在解码器中有一个相同的预测器,收到的预测误差信号经解量化器 DQ 以后,与预测器的输出相加,从而恢复出原信号。由于在编码端预测误差信号已被量化,因此在解码端所恢复出的信号 $y(n)$ 只是原信号 $x(n)$ 的近似值。

由于电视画面在内容上的连续性,空间域相邻像素之间有很强的相关性,可以利用这些相关性对当前的像素进行预测。这通常称为**空间域预测**。在解码端,根据前几个样值所得到的预测值 $\hat{x}(n)$ 已经是已知的,如果当前值 $x(n)$ 与 $\hat{x}(n)$ 相同[即 $e(n)=x(n)-\hat{x}(n)=0$],就不需要再传送新的数据给解码端了。因此,通过预测将 $x(n)$ 转换成 $e(n)$,在很大程度上降低了信息源的冗余度,这便是利用 DPCM 进行数据压缩的基本原理。

在图 3-14 中,X 表示被预测的像素,$x_1,x_2,\cdots,x_5$ 则是根据不同的预测方案有可能被选择用来对 X 进行预测的已知的像素。如果用作预测的样值与被预测的样值在同一行内(如图中的 x_1、x_2 与 X),称为**一维预测**;当用作预测的样值位于相邻的不同行上时(如图中 x_3、x_4 与 X)则称为**二维预测**。

一维预测利用了像素之间在水平方向上的相关性。对水平方向亮度变化缓慢的图像,有较好的预测效果。如果水平方向上亮度有突变,例如图 3-14 所示的图像是以 x_1 和 X 为界的黑白条,那么一阶的一维预测为

$$\hat{x}=x_1=0 \qquad \text{(黑电平)} \tag{3-35}$$

会给出错误的预测数值。在这种情况下,采用下面的二维预测给出较好的预测值,即

$$\hat{x}=x_1+\frac{x_4-x_5}{2}=\frac{1}{2} \tag{3-36}$$

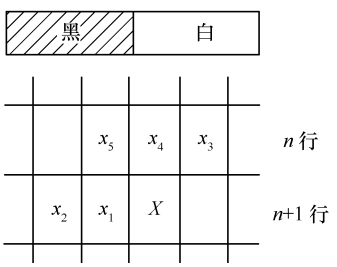


图 3-14　对应于图像黑白边界处的几个像素

下面考虑 N 阶预测器的设计问题。预测误差信号 $e(n)$ 平方值的统计平均由下式表示:

$$\langle e^2(n)\rangle = \langle [x(n)-\hat{x}(n)]^2\rangle = \langle [x(n)-\sum_{i=1}^{N} a_i x(n-i)]^2\rangle \tag{3-37}$$

式中〈 〉表示均值。显然，$\langle e^2(n)\rangle$最小时，表示在最小均方误差意义下，预测最准确，此时的预测器称为在最小均方误差 MMSE(Minimum Mean Square Error)意义下的**最佳预测器**。最佳预测器的系数 $a_i(i=1,\cdots,N)$可以通过如下求$\langle e^2(n)\rangle$极小值的方法获得：

令
$$\frac{\partial}{\partial a_i}\langle e^2(n)\rangle = 2\langle e(n)\frac{\partial}{\partial a_i}e(n)\rangle = 0 \qquad (i=1,2,\cdots,N) \tag{3-38}$$

$$\begin{aligned}\langle e(n)\frac{\partial}{\partial a_i}e(n)\rangle &= \langle [x(n)-\sum_{k=1}^{N} a_k x(n-k)][-x(n-i)]\rangle \\ &= -\langle x(n)x(n-i)\rangle + \sum_{k=1}^{N} a_k\langle x(n-k)x(n-i)\rangle \\ &= 0 \qquad (i=1,2,\cdots,N)\end{aligned} \tag{3-39}$$

将上式用输入序列的自相关函数 R 来表示：

$$R(i)-\sum_{k=1}^{N} a_k R(i-k) = 0 \qquad (i=1,2,\cdots,N) \tag{3-40}$$

如果对需要压缩的某类图像的自相关函数已经做过测量的话，则可通过求解(3-40)式所表示的方程组，获得最佳预测器的系数值。此时的均方误差值则为

$$\langle e^2(n)\rangle_{\min} = \langle e(n)[x(n)-\sum_{k=1}^{N} a_k x(n-k)]\rangle \tag{3-41}$$

由(3-39)式可知，

$$\langle e(n)x(n-i)\rangle = -\langle e(n)\frac{\partial}{\partial a_i}e(n)\rangle = 0 \tag{3-42}$$

于是(3-41)式可简化为

$$\langle e^2(n)\rangle_{\min} = \langle e(n)x(n)\rangle = R(0)-\sum_{k=1}^{N} a_k R(k) \tag{3-43}$$

由上式可以看出，预测误差的平均功率比原信号的功率 $R(0)$ 要小。在相同的均方量化误差下，$e(n)$ 比 $x(n)$ 要求较少的量化级数，因此，传送 $e(n)$ 比传送 $x(n)$ 的数据率要低。

在图 3-13 所示的电路中，量化器处于预测差分环路之外。编码端利用原始信号为参考进行预测，而在解码端却是以量化后的信号为参考进行预测的，这种两端参考信号不相同的情况，称为**失配**，它导致重建信号 $y(n)$与输入信号 $x(n)$ 之间存在较大误差。图 3-15 给出了一个严格意义上的 DPCM 原理方框图，图中编码器的反馈环路模拟了解码端的结构：量化后的预测误差信号经解量化器后，与预测值相加构成预测器的输入信号，从而使解码端重建信号的误差降低。

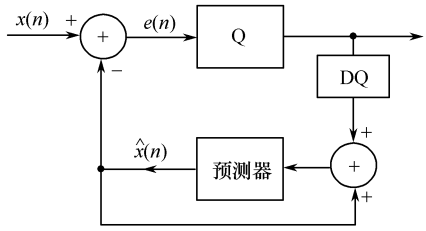


图 3-15　量化器处于预测环路之内的预测编码电路

3.5.2　序列图像中运动矢量的估值

1. 运动矢量估值的必要性

电视图像是由在时间上相互间隔为帧周期(1/25s、或 1/30s)的一帧帧图像组成的图像序列。可以想象，在拍摄同一场景时，时间上相邻的两幅图像之间在内容上差异不会太大，或者说后一帧的内容与前一帧重复的部分很多。用数学的术语来讲，二者是相关的。消除序列图像在时间上的相关性(冗余度)，是视频信号压缩编码的一条重要途径。

序列图像在时间上的冗余情况可分为如下几种：① 对于静止不动的场景，当前帧和前一帧的

图像内容是完全相同的；② 对于运动的物体，只要知道其运动规律，就可以从前一帧图像推算出它在当前帧中的位置来；③ 摄像镜头对着场景进行横向移动（称为滑镜头）、焦距变化等操作会引起整个图像的平移、放大或缩小，对于这种情况，只要摄像机的运动规律和镜头改变的参数已知，图像随时间所产生的变化也是可以推算出来的。由电视图像的这些特点我们可以想到，发送端不一定必须把每帧图像上所有的像素都传给接收端，而只要将物体（或摄像机）的运动信息告知收端，收端就可根据运动信息和前一帧图像的内容来更新当前帧图像，这比全部传送每帧图像的具体细节所需的数据量小得多。

要这样做，一个首先需要解决的问题是如何从序列图像中提取出有关物体运动的信息，这通常称为**运动估值**。本节讨论的内容就是运动估值的方法。摄像机的运动和参数变化影响到图像的每个像素，我们将这种由摄像机运动导致的像素位置变化称为**全局运动**。本节将不涉及全局运动估值问题，有兴趣的读者可以参考其他文献[5]。

为了简单起见，本节介绍的物体运动估值方法基于如下的假设：

（1）物体是刚体，且只在与摄像机镜头的光轴（即穿过镜头球面中心和球心的轴）垂直的平面内移动。这就是说，物体的形变、旋转、镜头焦距的变更等因素不考虑在内；

（2）无论物体移动到任何位置，照明条件都不变，也就是说，同一物体在所有序列图像中亮度没有变化；

（3）被物体遮挡的背景和由于物体移开而新暴露出来的背景部分不做特殊考虑。在上述假设下，t 时刻运动物体的像素 b_t 可以用它在时间 τ 以前的值 $b_{t-\tau}$ 表示出来，即

$$b_t(\boldsymbol{z}) = b_{t-\tau}(\boldsymbol{z} - \boldsymbol{D}) \tag{3-44}$$

式中，b 代表像素亮度值；$\boldsymbol{z}=(x,y)^{\mathrm{T}}$ 为位置矢量，T 代表矢量的转置；$\boldsymbol{D}$ 为在时间间隔 τ 内物体运动的**位移矢量**（Displacement Vector）。严格地讲，位移矢量与**运动矢量**（Motion Vector，MV），在概念上是有区别的，因为位移只是物体运动的一种方式，但由于在当前技术条件下，位移几乎是进行视频数据压缩时所考虑的唯一运动方式，因此在相关的文献中，常将 $\boldsymbol{D}$ 称为运动矢量。上式表示 t 时刻的图像是 $(t-\tau)$ 时刻的图像经适当移位后的结果。因此通过比较相距时间为 τ 的两帧图像可以估计出在这段时间间隔内物体的位移 $\boldsymbol{D}$。根据这一公式进行运动估值的方法主要分为两大类[2,6]，分别称为**块匹配方法**和**像素递归方法**。由于块匹配方法是目前视频压缩编码广泛采用的方法，因此本节仅针对它进行讨论。

2. 块匹配方法

按照一般的想法，运动估值应当首先将图像中的静止背景和运动物体区分开来，然后对运动物体的实际位移进行估值，但是从图像中分割出物体常常是一个困难的任务。为了回避这个困难，在块匹配方法中，将图像划分为许多互不重叠的块（如 16×16），并假设块内所有像素的位移量都相同。这实际上意味着将每个块视为一个“运动物体”。假设在图像序列中，t 时刻对应于第 k 帧图像，$t-\tau$ 时刻对应于 $k-1$ 帧图像。对于 k 帧中的一个块，在 $k-1$ 帧中找与其最相似的块，称为**匹配块**，并认为该匹配块在 $k-1$ 帧中的位置，就是 k 帧块位移前的位置，根据(3-44)式则可以得到该块的位移矢量 $\boldsymbol{D}$。此时，$k-1$ 帧称为 k 帧的**参考帧**。

为了节省计算量，在 $k-1$ 帧中的匹配搜索只在一定范围内进行。假设在 τ 时间间隔内块的最大可能水平和垂直位移量为 d_m 个像素，则搜索范围 SR 为

$$SR = (M+2d_m)\times(N+2d_m) \tag{3-45}$$

式中，M、N 分别为块在水平和垂直方向上的像素数。图 3-16 给出了块与搜索范围的相对位置关系。显然，在块匹配方法中最重要的两个问题是：判别两个块匹配的准则和寻找匹配块的搜索方法。

对这两个问题的不同解决方案构成了不同的算法。

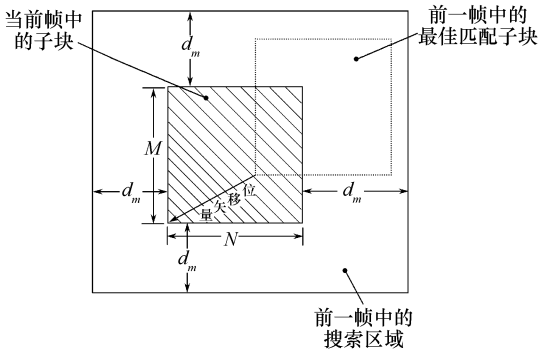


图 3-16　块与搜索范围 SR 的位置关系

判断两个块相似程度的最直接的准则是归一化的二维互相关函数 NCCF，其定义为

$$\mathrm{NCCF}(i,j)=\frac{\sum_{m=1}^{M}\sum_{n=1}^{N}b_k(m,\ n)b_{k-1}(m+i,\ n+j)}{\left[\sum_{m=1}^{M}\sum_{n=1}^{N}b_k^2(m,\ n)\right]^{1/2}\left[\sum_{m=1}^{M}\sum_{n=1}^{N}b_{k-1}^2(m+i,\ n+j)\right]^{1/2}} \tag{3-46}$$

式中的时间和位置已用相应的离散量表示，分子为在第 k 帧中的块与在 $k-1$ 帧中与该块对应位置相差 i 行、j 列的块之间的互相关函数，分母中括号里的项分别代表这两个块各自的自相关函数的峰值。当 NCCF 为最大值时两个块匹配，此时对应的 i、j 值即构成位移矢量 $\boldsymbol{D}$。

在实际应用中，常常使用如下计算比较简单的判断块匹配的准则：

(1) 块亮度的均方差值(MSE)

$$\mathrm{MSE}(i,\ j)=\frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}\left[b_k(m,\ n)-b_{k-1}(m+i,\ n+j)\right]^2$$
$$(-d_m\leqslant i,\ j\leqslant d_m) \tag{3-47}$$

(2) 块亮度差的绝对值均值(MAD)

$$\mathrm{MAD}(i,\ j)=\frac{1}{MN}\sum_{m=1}^{M}\sum_{n=1}^{N}\left|b_k(m,\ n)-b_{k-1}(m+i,\ n+j)\right|$$
$$(-d_m\leqslant i,\ j\leqslant d_m) \tag{3-48}$$

(3) 块亮度差的绝对值和(SAD)

$$\mathrm{SAD}(i,j)=MN\cdot\mathrm{MAD}(i,j) \tag{3-49}$$

当 MSE 或 MAD 或 SAD 最小时，表示两个块匹配。

研究结果表明，匹配判别准则的不同对匹配的精度，也就是对位移矢量估值的精度影响不大。因此，(3-49)式所表示的不含有乘法和除法的 SAD 准则成为最常使用的匹配判别准则。

为了寻找最佳的匹配块，我们需要将 $k-1$ 帧中对应的块在整个搜索区内沿水平和垂直方向逐个像素移动，每移动一次计算一次判决函数(如 SAD)。总的移动次数(搜索点)Q 为

$$Q=(2\ d_m+1)^2 \tag{3-50}$$

这种搜索方式称为**全搜索**。全搜索的运算量是相当大的。使用 SAD 为准则时每个像素要进行三个基本运算(相减、求绝对值、求和)，对一个块进行全搜索要求 $3\times(2d_m+1)^2\times MN$ 次运算。假设视频图像的分辨率为 $N_\mathrm{W}\times N_\mathrm{H}$，则每帧有 $(N_\mathrm{W}\times N_\mathrm{H})/MN$ 个块，即使在 $d_m=7$、$[N_\mathrm{W},N_\mathrm{H}]=[352,288]$和帧率 $f=30$ 时，所需的运算量也达到 2.05 千万次/秒。一般来说，运动估值的运算量通常占到现行标准下视频压缩编码的 60%～80%。

3. 块匹配的快速搜索方法

为了加快搜索过程，人们提出了许多不同的搜索方法。图 3-17 所示的**三步法**是早期提出的一

个典型快速搜索算法。它通过降低搜索点的数目来降低算法的计算复杂度。其搜索过程为：第一步，以待匹配块中心的**同位**像素（即前一帧中与之位置相同的像素）为中心，在中心点和与其相距 4 个像素的 8 个邻域点上计算判决函数 SAD 值，取 SAD 值最小的点作为下一步搜索的中心；第二步，以该点为中心，对与中心相距 2 个像素的未搜索过的邻域点进行搜索；第三步，以上一步中 SAD 值最小的点为中心，对距离中心 1 个像素的未搜索过的邻域点进行搜索，最终找到最佳匹配位置。图中带有数字的圆圈代表每一步骤中的搜索点，带有数字的方块代表该步骤中 SAD 值最小的位置。本例中，在 $d_m \leqslant 7$ 的搜索范围内，通过三步找到最佳匹配位置（$i+7, j-2$）。早期提出的其他算法，如二维对数法、共轭方向法等[2,5,6]，与三步法相类似，只是每个步骤搜索点所构成的图形的**形状**和**大小**不同。

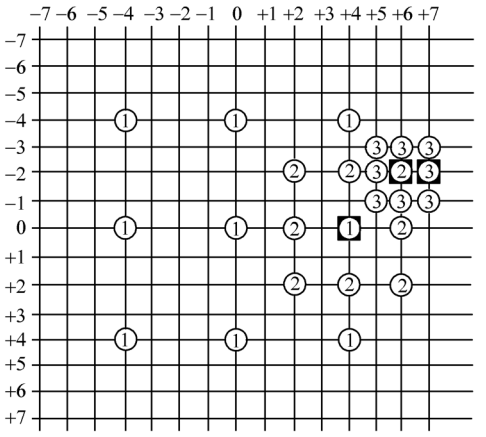


图 3-17　三步法的搜索过程

几乎所有的快速搜索算法都基于如下的假设：当偏离最佳匹配位置（运动矢量 $\boldsymbol{D}=\boldsymbol{D}_{\text{opt}}$）时，判决函数（匹配误差）值是单调上升的。因此无需搜索所有的点，只要沿着误差值减少的方向进行搜索，就能找到最佳匹配位置。图 3-18 给出了一维情况下沿误差曲线搜索的情况（如图实线上箭头所示）。但是当误差曲线非凸时（如图中虚线示），这种搜索方法则可能落入局部极值点，例如从图中 A 点左侧任意点沿误差减小方向的搜索将中止于 A 点，而不能搜索到最佳匹配位置（全局极小点）。

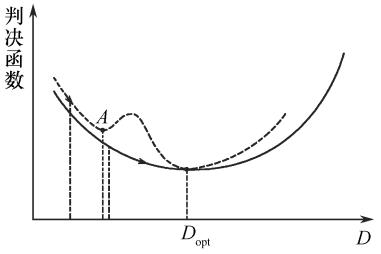


图 3-18　运动矢量估值过程示意图

保证在任何情况下找到全局极值点是一个困难的问题，但是从另一个角度来考虑，在全局极值点的邻近区域，误差曲线（二维时为曲面）为凸的假设总是成立的，因此只要有一个搜索点落在全局极值点的附近，搜索到最佳匹配位置的概率就会很大。

根据上述想法，近年来人们提出了许多新的快速搜索算法，具有代表性的有 MVFAST，APDZS 和 EPZS 等[7~9]，它们不仅大大提高了搜索的速度，而且具有与全搜索方法相当的性能。这些算法的基本策略可以概括如下。

（1）运动矢量预测

由于图像内容的连续性，相邻块的运动矢量一般是相近的，换句话说，运动矢量场（运动矢量的分布函数）在空间和时间上一般是平滑、渐变的，因此，我们可以根据已进行过运动估值的相邻块的运动矢量（假设它们的估值是正确的），对当前待匹配块的运动矢量进行预测，然后从预测位置开始进行搜索。如果预测得足够准确，那么预测点落在误差曲线的全局极值点附近，这不仅能够缩短搜索需要的时间，而且使搜索到全局极值点的概率增高。

在图 3-19 中，设 k 帧中灰色块为当前待匹配的块，它的 MV 的预测值可以选择为：(0,0)，或当前块的左、上和右上三个空间邻域块的运动矢量 $\boldsymbol{MV}_{\mathrm{L}}$，$\boldsymbol{MV}_{\mathrm{T}}$ 和 $\boldsymbol{MV}_{\mathrm{TR}}$，也可以选择为这三个 MV 的中值 $\boldsymbol{MV}_{\mathrm{med}}$，或前一帧同位块的运动矢量 $\boldsymbol{MV}_{\mathrm{C}}$。其他预测值还可以考虑：前一帧同位块的上、下、左、右 4 个邻域块的 MV、前 2 帧的“加速度”$\boldsymbol{MV}=\boldsymbol{MV}_{\mathrm{C}}+(\boldsymbol{MV}_{\mathrm{C}}-\boldsymbol{MV}_{\mathrm{C2}})$（见图 3-19），以及上述各预测值的线性组合等。我们将各个 MV 预测值所确定的一组点作为搜索的候选起始点，计算它们的 SAD 值，并选择 SAD 值最小的点作为最佳起始点；然后以该点为中心类似于三步法那样，计算规定的搜索图形上各搜索点的 SAD 值，并按 SAD 值减小的方向移动搜索图形，直至找到最佳匹配位置为止。由于搜索采用了多个候选起始点，所以使得因 MV 预测不准确而引起的起始点远离全局极值点的风险降低。

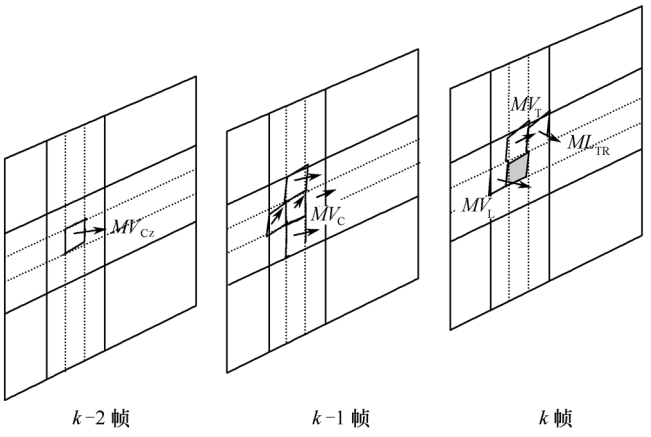


图 3-19　运动矢量的预测

(2) 搜索提前中止

所谓“提前中止”是指，在某个搜索点上如果匹配误差 SAD 小于预先设定的阈值 T，则认为匹配已经足够好，搜索中止。人们通过研究发现，大多数图像块的运动矢量与空间邻块运动矢量的中值 $\boldsymbol{MV}_{\mathrm{med}}$ 相关性最强，与(0,0)、$\boldsymbol{MV}_{\mathrm{L}}$、$\boldsymbol{MV}_{\mathrm{T}}$、$\boldsymbol{MV}_{\mathrm{TR}}$ 和 $\boldsymbol{MV}_{\mathrm{C}}$ 的相关性次之，与其他 MV 预测值的相关性则小一些。显然，先搜索相关性强的预测点，可能较早满足提前中止的条件，从而提高搜索效率。

提前中止阈值 T 的选取是一个值得考虑的问题，阈值过小达不到提前中止的目的；过大则搜索不到最佳匹配位置。一般来说，针对不同情况可以设置不同的阈值，例如对相关性强的点，阈值严格一些，对相关性弱的点则宽松一些。除了固定阈值之外，还可以动态地选取阈值。考虑到待匹配块的 SAD 值与它的空间邻域（左、上和上右）和时间邻域（同位）块的 SAD 值之间也存在着相关性，一种动态阈值的选择方法为

$$T = a \cdot \min(\mathrm{SAD}_{\mathrm{L}}, \mathrm{SAD}_{\mathrm{T}}, \mathrm{SAD}_{\mathrm{TR}}, \mathrm{SAD}_{\mathrm{c}}) + b \tag{3-51}$$

式中，a、b 为预定的常数。

(3) 紧凑的搜索图形

在三步法中，为了提高搜索速度，在开始时采用了大尺寸（隔 4 个像素）的矩形搜索图形；在搜索趋向于匹配点的过程中，搜索图形的尺寸逐步减小。由于近年提出的快速算法都进行了不同程度的运动矢量预测，搜索的起始点已经接近于匹配点，因此可以采用紧凑的搜索图形。图 3-20给出了几个例子，其中(a)称为“大钻石”，在距离匹配点稍远处使用；(b)为“小钻石”，在匹配点附近使用。判断当前搜索点距离匹配点的远近可以有多种准则，一种常用的办法是检查搜索点 SAD 值的大小。SAD 值小意味着接近匹配点，可以使用小尺寸图形。

上述三种基本策略的不同运用形成了不同的算法。图 3-21 给出了一个搜索过程的示例，其中

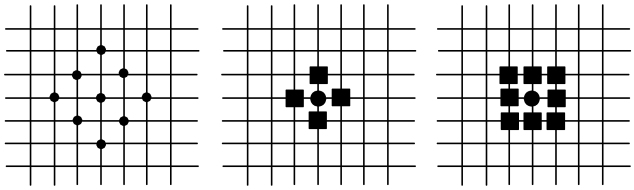


图 3-20　搜索图形

带有数字的圆圈代表每一个步骤的搜索点，带有数字的方块代表该步骤中的 SAD 值最小的位置（见习题 9）。搜索图形为小钻石。

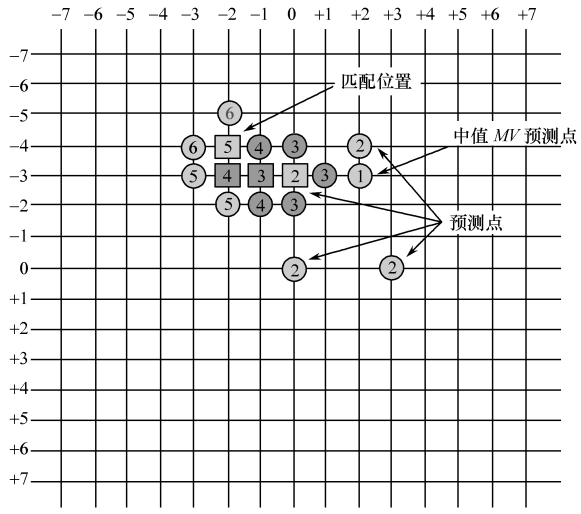


图 3-21　搜索过程示例

3.5.3　具有运动补偿的帧间预测

1. 前向预测

与消除图像中相邻像素间的空间冗余度一样，消除序列图像在时间上的相关性，也可以采用预测编码的办法，即不直接传送当前帧（即 k 帧）的像素值 x[见图 3-22(a)]，而传送 x 与前一帧（即 $k-1$ 帧）同位像素 x' 之间的差值，这称为**帧间预测**。对隔行扫描的电视信号，也可以用前一场来预测当前场的像素（场间预测）。当图像中存在着运动物体时，简单的预测不能收到好的效果。例如图 3-22(b)中，当前帧与前一帧的背景完全一样，只是小球平移了一个位置。如果简单地以 $k-1$ 帧的同位像素值作为 k 帧像素的预测值，则在实线和虚线所示的圆内预测误差都不为零。如果已经知道了小球的位移矢量，可以从小球在 $k-1$ 帧的位置推算出它在 k 帧中的位置来，而背景图像（不考虑被遮挡的部分）仍以前一帧的背景代替。将这种考虑了小球位移的 $k-1$ 帧图像作为 k 帧的预测值，就比简单的预测准确得多，从而可以达到更高的数据压缩比。这种预测方法称为**具有运动补偿的帧间预测**。

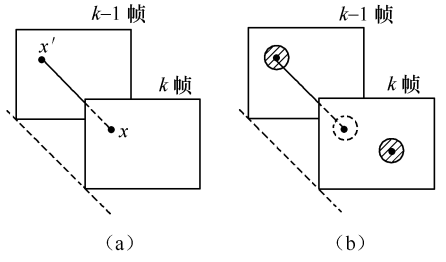


图 3-22　帧间预测与具有运动补偿的帧间预测

从原理上讲，具有运动补偿的帧间预测应包括如下几个基本步骤：

(1) 将图像分割成静止的背景和若干运动的物体，各个物体可能有不同的位移，但构成同一物体的所有像素的位移相同。通过运动估值得到每个物体的位移矢量；

(2) 利用位移矢量计算经运动补偿后的预测值；

(3) 除了对预测误差进行编码、传送以外，还需要传送位移矢量以及如何进行运动物体和静止背景分割等方面的附加信息。

图 3-23 给出了这种预测器的原理方框图。图中向下的虚线箭头标出分割的结果送进运动估值单元，以便针对不同物体进行位移估值。而向上的虚线箭头表示用估值的结果和分割的结果一起去控制预测器，以得到经运动补偿后的预测图像。编码器的最终输出为帧间预测误差、位移矢量和分割产生的地址信息等。

如前所述，将图像分割成静止区域和不同的运动区域是一项困难的工作，当要求实时地完成这项运算时就更加困难。一种简化的办法是将图像分割成块，每块看成是一个物体，按 3.5.2 节中讲的块匹配的方法估计每个块的位移矢量，将经过位移补偿的帧间预测误差(Displaced Frame Difference，DFD)和位移矢量 $\boldsymbol{D}$ 传送给接收端，接收端就可以按下式从已经收到的前一帧信息中恢复出该块：

$$b_k(\boldsymbol{z}) = b_{k-1}(\boldsymbol{z} - \boldsymbol{D}) + \mathrm{DFD}(\boldsymbol{z}, \boldsymbol{D}) \qquad (3\text{-}52)$$

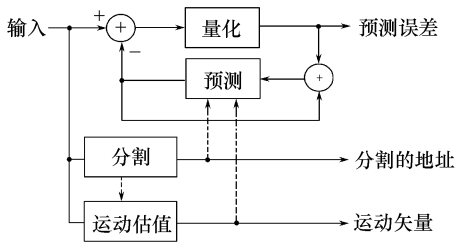


图 3-23 具有运动补偿的帧间预测器功能方框图

图 3-24 中表示出了 k 帧各块及它们在 $k-1$ 帧中对应的匹配块之间的关系。从该块的预测误差和它的位移矢量所指向的 $k-1$ 帧中的匹配块，可以恢复出 k 帧中的块来。显然，当一个块中的像素实际上属于位移量不同的物体时，这种对整个块用同一位移作的预测则不够准确，这将使预测误差 DFD 增加，从而影响数据压缩比的提高。

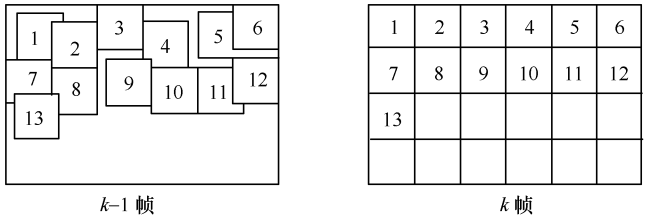


图 3-24 块匹配方式的帧间预测

2. 后向预测和双向预测

上面介绍的用 $k-1$ 帧预测 k 帧图像的预测方式称为**前向预测**。如果待预测的块是在 k 帧，而搜索区域处于 $k+1$ 帧之内，也就是从后续的 $k+1$ 帧图像预测前面的 k 帧图像，这种方式称为**后向预测**。对于在 $k-1$ 帧中被覆盖，而从 k 帧开始新暴露出的物体(或背景)，使用后向预测可以得到更好的预测值。

第三种预测方式是采用前、后两帧来预测中间帧，这称为**双向预测**。如图 3-25 所示，对于 k 帧中的块(灰色块)，先从 $k-1$ 帧中找到它的最佳匹配块，从而得到该块从 $k-1$ 到 k 帧的位移矢量 $\boldsymbol{D}_1$，再利用后向预测得到它从 $k+1$ 到 k 帧的位移矢量 $\boldsymbol{D}_2$，然后将经过运动补偿的前向预测值或后向预测值，或二者平均值作为 k 帧块的预测值(视哪种预测误差最小而定)。这样的做法与单纯的前向预测相比，可以进

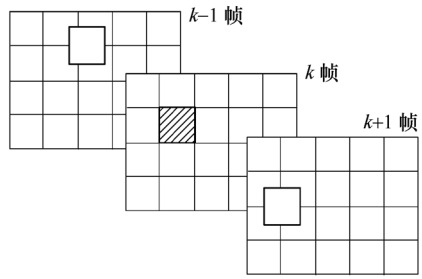


图 3-25 双向预测

一步降低预测误差，从而提高数据压缩比。

双向预测所付出的代价是，对每一个图像块需要传送两个位移矢量给接收端，而且 k 帧的恢复必须等到 $k+1$ 帧解码之后才能进行。也就是说，输入序列的帧顺序是 $k-1$、k、$k+1$，编码和解码运算的帧顺序是 $k-1$、$k+1$、k，而图像显示的顺序又是 $k-1$、k、$k+1$。要保持处理和显示的连续性，在编码端和解码端分别需要引入一帧的延时。

3.6 正交变换编码

3.6.1 最佳线性正交变换

正交变换编码与预测编码一样，都是利用去除信源序列的相关性来达到数据压缩的目的。它们之间的不同之处在于，预测编码是在空间域或时间域内进行，而变换编码则是在变换域（频率域）内进行的。

在数据压缩中，进行正交变换的目的是希望为给定的某类信号找到一种最有效的表示方式。假定一个离散时间（或空间）信号由 N 个采样值所组成，则可以认为它是一个在 N 维空间中的点，而每个采样值代表 N 维信号空间中数据向量 $\boldsymbol{X}$ 的一个分量。为了找到有效的表示方法，我们选取 $\boldsymbol{X}$ 的一个正交变换，使

$$\boldsymbol{Y} = \boldsymbol{T}\boldsymbol{X} \tag{3-53}$$

式中，$\boldsymbol{Y}$ 与 $\boldsymbol{T}$ 分别为变换向量和正交变换矩阵。我们的目的是要寻找一个变换矩阵 $\boldsymbol{T}$，将经上式变换得到的 $\boldsymbol{Y}$ 用一个由 $M(M<N)$ 个分量构成的子集来近似，当删去 $\boldsymbol{Y}$ 中剩下的 $(N-M)$ 个分量，仅用这个子集来恢复 $\boldsymbol{X}$ 时，不引起明显的误差。这样就可以用 $\boldsymbol{Y}$ 的这个只有 M 个分量的子集来代表有 N 个分量的信号 $\boldsymbol{X}$，从而达到数据压缩的目的。

为了找到“最佳”的正交变换，通常使用的判断准则是，能使得在恢复 $\boldsymbol{X}$ 时所产生的均方误差最小。

设变换矩阵具有如下形式：

$$\boldsymbol{T}^{\mathrm{H}} \equiv \boldsymbol{T}^{*\mathrm{T}} = [\boldsymbol{\Phi}_1, \boldsymbol{\Phi}_2, \cdots, \boldsymbol{\Phi}_N] \tag{3-54}$$

式中，上标 $*$ 和上标 T 分别代表矩阵的共轭与转置，上标 H 代表共轭、且转置，$\boldsymbol{\Phi}_i$ 是 N 维的列向量，且是**标准正交**（Orthonormal）的，即

$$\boldsymbol{\Phi}_i^{\mathrm{H}} \cdot \boldsymbol{\Phi}_k = \begin{cases} 1 & (i = k) \\ 0 & (i \neq k) \end{cases} \tag{3-55}$$

由于 $\boldsymbol{\Phi}_i$ 相互正交，所以它们是线性独立的，即它们之中的任何一个不能由其余向量的线性组合来产生。我们知道，N 个线性独立的向量可以生成一个 N 维空间，这一组向量称为该空间的**基**，其中的每一个 $\boldsymbol{\Phi}_i$ 称为**基向量**。例如，三维欧几里得空间的三个基向量分别是 $\boldsymbol{i}$，$\boldsymbol{j}$ 和 $\boldsymbol{k}$。

由(3-54)式和(3-55)式，得到 $\boldsymbol{T}\boldsymbol{T}^{\mathrm{H}} = \boldsymbol{T}^{\mathrm{H}}\boldsymbol{T} = \boldsymbol{I}$，即 $\boldsymbol{T}$ 的逆矩阵为

$$\boldsymbol{T}^{-1} = \boldsymbol{T}^{\mathrm{H}} \tag{3-56}$$

满足上述条件的矩阵称为**酉（Unitary）矩阵**，它所对应的变换称为**酉变换**。当 $\boldsymbol{T}$ 为实数矩阵，且 $\boldsymbol{T}^{-1} = \boldsymbol{T}^{\mathrm{T}}$ 时，$\boldsymbol{T}$ 称为**正交矩阵**，其对应的变换称为**正交变换**。数据压缩中所研究的主要是正交（或酉）变换。我们熟悉的傅氏变换，当其基向量 $\boldsymbol{\Phi}_i = \frac{1}{\sqrt{N}}\left[\mathrm{e}^{\mathrm{j}\frac{2\pi}{N}i\cdot 0}, \mathrm{e}^{\mathrm{j}\frac{2\pi}{N}i\cdot 1}, \cdots, \mathrm{e}^{\mathrm{j}\frac{2\pi}{N}i(N-1)}\right]^{\mathrm{T}}$ 时，就是一个酉变换。

由(3-53)式和(3-56)式，得到

$$\boldsymbol{X} = \boldsymbol{T}^{\mathrm{H}}\boldsymbol{Y} = y_1\boldsymbol{\Phi}_1 + y_2\boldsymbol{\Phi}_2 + \cdots + y_N\boldsymbol{\Phi}_N = \sum_{i=1}^{N} y_i\boldsymbol{\Phi}_i \tag{3-57}$$

上式表明将 $\boldsymbol{X}$ 转换到由基向量 $\boldsymbol{\Phi}_i(i=1,\cdots,N)$ 生成的 N 维空间（通常称为**变换域**）中，y_i 代表 $\boldsymbol{X}$ 在 $\boldsymbol{\Phi}_i$ 上投影的大小，称为**变换系数**。因此，由变换系数所构成的向量 $\boldsymbol{Y}$ 是信号 $\boldsymbol{X}$ 在变换域中的表示。

假设信号 $\boldsymbol{X}$ 是一个均值为零的随机向量，即$\langle \boldsymbol{X} \rangle = 0$。若只保留 $M(M<N)$个变换系数，将其余$(N-M)$个系数置为零，则所得到的 $\boldsymbol{X}$ 的近似值 $\boldsymbol{X}_M$ 与原信号的差值 $\Delta \boldsymbol{X}$ 为

$$\Delta \boldsymbol{X} = \boldsymbol{X} - \boldsymbol{X}_M = \sum_{i=M+1}^{N} y_i \boldsymbol{\Phi}_i \tag{3-58}$$

其均方误差 MSE 则为

$$\begin{aligned} \mathrm{MSE} &= \langle \| \Delta \boldsymbol{X} \|^2 \rangle = \langle \Delta \boldsymbol{X}^{\mathrm{H}} \cdot \Delta \boldsymbol{X} \rangle = \langle \sum_{i=M+1}^{N} \sum_{k=M+1}^{N} y_i^* \cdot y_k \boldsymbol{\Phi}_i^{\mathrm{H}} \cdot \boldsymbol{\Phi}_k \rangle \\ &= \sum_{i=M+1}^{N} \langle | y_i |^2 \rangle \end{aligned} \tag{3-59}$$

上式的最后一步由基向量的正交性得到。由(3-53)式可知，$y_i = \boldsymbol{\Phi}_i^{\mathrm{H}} \cdot \boldsymbol{X}$，且标量的转置为其自身，则(3-59)式可改写为

$$\mathrm{MSE} = \sum_{i=M+1}^{N} \langle y_i \cdot y_i^{\mathrm{H}} \rangle = \sum_{i=M+1}^{N} \boldsymbol{\Phi}_i^{\mathrm{H}} \langle \boldsymbol{X} \cdot \boldsymbol{X}^{\mathrm{H}} \rangle \boldsymbol{\Phi}_i \tag{3-60}$$

当$\langle \boldsymbol{X} \rangle = 0$ 时，$\langle \boldsymbol{X} \cdot \boldsymbol{X}^{\mathrm{H}} \rangle$即为 X 的协方差矩阵 $\boldsymbol{\Sigma}_X$ 。

利用拉格朗日乘数法可以证明[10]，在 $\boldsymbol{\Phi}_i^{\mathrm{H}} \cdot \boldsymbol{\Phi}_i = 1$ 的条件下，使 MSE 为最小的条件是

$$\boldsymbol{\Sigma}_X \Phi_i = \lambda_i \boldsymbol{\Phi}_i \qquad (1 \leqslant i \leqslant N) \tag{3-61}$$

由上式看出 $\boldsymbol{\Phi}_i$ 和λ_i 分别是矩阵 $\boldsymbol{\Sigma}_X$ 的本征向量和本征值。这也就是说，以信号的协方差矩阵 $\boldsymbol{\Sigma}_X$ 的本征向量$\boldsymbol{\Phi}_i (i = 1, 2, \cdots, N)$ 组成的变换矩阵是均方误差最小准则下的最佳变换矩阵，用此矩阵构成的最佳变换 $\boldsymbol{Y} = \boldsymbol{T}\boldsymbol{X}$ 称为卡南—洛伊夫变换(Karhunen-Loeve Transform)，简称 KLT。

经 KL 变换后，Y 的协方差矩阵 $\boldsymbol{\Sigma}_Y$ 为

$$\boldsymbol{\Sigma}_Y = < \boldsymbol{Y} \cdot \boldsymbol{Y}^{\mathrm{H}} > = \boldsymbol{T} \boldsymbol{\Sigma}_X \boldsymbol{T}^{\mathrm{H}} = \begin{bmatrix} \lambda_1 & & & \\ & \lambda_2 & & 0 \\ & & \ddots & \\ & 0 & & \\ & & & \lambda_N \end{bmatrix} \tag{3-62}$$

根据协方差矩阵的定义可知，$\boldsymbol{\Sigma}_Y$ 矩阵对角线上的元素为 $\boldsymbol{Y}$ 的分量 y_i 的方差，而非对角线上的元素代表 $\boldsymbol{Y}$ 的分量之间的协方差值。(3-62)式说明，KL 变换解除了随机向量 $\boldsymbol{X}$ 的分量之间的相关性，在变换域中 $\boldsymbol{Y}$ 的各分量之间是互不相关的(因为协方差为零)，从而有利于对各分量分别独立地进行处理。同时，将(3-61)式代入(3-60)式，得到

$$\mathrm{MSE}_{\min} = \sum_{i=M+1}^{N} \lambda_i \tag{3-63}$$

上式说明最小均方误差等于 $\boldsymbol{Y}$ 向量中被丢弃的分量的方差之和。由此可知，应该选择具有较大方差的 M 个 $\boldsymbol{Y}$ 分量所构成的子集来恢复 $\boldsymbol{X}$，以使得恢复后所产生的误差最小(能量损失最小)。

由于构成 KL 变换的基向量是 $\boldsymbol{X}$ 协方差矩阵的本征向量，因此，KLT 的基向量与信号的统计特性有关，必须针对某一类信号具体地设计。这一点是与傅氏变换不同的。在傅氏变换中，不管对具有何种特性的信号，均使用同样的基向量。此外，KLT 也缺乏相应的快速算法，因此在数据压缩中应用并不普遍，但它常被用来作为评价各种变换的性能的基准。

3.6.2 离散余弦变换

选择不同的正交基向量，可以得到不同的正交变换。从数学上可以证明，各种正交变换都能在不同程度上减小随机向量各分量之间的相关性[10]，而且信号经过大多数正交变换后，能量会相对集中在少数变换系数上，删去对信号贡献小(方差小)的系数，只利用保留下来的系数恢复信号时，不会引起明显的失真。因此，不同的正交变换，例如，离散傅氏变换(DFT)，离散余弦变换(DCT)，

哈尔变换(HT),沃尔什—哈达马变换(WHT)等均在数据压缩中得到应用,只是在均方误差准则下,性能不如 KLT 好。

当信号的统计特性符合一阶平稳马尔柯夫过程,而且相关系数接近于 1 时(许多图像信号都可以足够精确地用此模型描述),DCT 十分接近于信号的最佳变换 KLT[10],变换后的能量集中程度较高。即使信号的统计特性偏离这一模型,DCT 的性能下降也不显著。由于 DCT 的这一特性,再加上其基向量是固定的,并具有快速算法等原因,它在图像和视频数据压缩中得到了广泛的应用。

顾名思义,DCT 的基向量由余弦函数构成。一维 DCT 的正变换和反变换分别由下式定义:

$$S(n) = \left(\frac{2}{N}\right)^{1/2} C(n) \sum_{k=0}^{N-1} s(k) \cos \frac{(2k+1)n\pi}{2N} \qquad (n = 0,1,\cdots,N-1) \tag{3-64}$$

$$s(k) = \left(\frac{2}{N}\right)^{1/2} \sum_{n=0}^{N-1} C(n) S(n) \cos \frac{(2k+1)n\pi}{2N} \qquad (k = 0,1,\cdots,N-1) \tag{3-65}$$

式中,$s(k)$ 为信号样值;$S(n)$ 为变换系数,且

$$C(n) = \begin{cases} \dfrac{1}{\sqrt{2}} & n = 0 \\ 1 & n \neq 0 \end{cases} \tag{3-66}$$

由一维的 DCT 可以直接扩展到二维,即

$$S(u,v) = \frac{2}{N} C(u) C(v) \sum_{j=0}^{N-1} \sum_{k=0}^{N-1} s(j,k) \cos \frac{(2j+1)u\pi}{2N} \cos \frac{(2k+1)v\pi}{2N}$$
$$(u = 0,1,\cdots,N-1; v = 0,1,\cdots,N-1) \tag{3-67}$$

$$s(j,k) = \frac{2}{N} \sum_{u=0}^{N-1} \sum_{v=0}^{N-1} C(u) C(v) S(u,v) \cos \frac{(2j+1)u\pi}{2N} \cos \frac{(2k+1)v\pi}{2N}$$
$$(j = 0,1,\cdots,N-1; k = 0,1,\cdots,N-1) \tag{3-68}$$

其中

$$C(u) = \begin{cases} \dfrac{1}{\sqrt{2}} & u = 0 \\ 1 & u \neq 0 \end{cases} \tag{3-69}$$

$$C(v) = \begin{cases} \dfrac{1}{\sqrt{2}} & v = 0 \\ 1 & v \neq 0 \end{cases} \tag{3-70}$$

图 3-26(a)表示出了二维 DCT 的基函数,(b)表示了 $N=8$ 时的信号矩阵和变换系数矩阵。

DCT 与 DFT 一样,如果没有快速算法,就很难在实际中得到应用。如果直接利用正变换公式进行一个一维 N 点 DCT 的计算,需要作 N^2 次乘法运算和 $N(N-1)$次加法运算,与直接计算的 DFT 运算量相同。与 FFT 相类似,根据基函数的周期性和对称性,人们已经提出许多种快速 DCT 算法,典型的快速算法请参见本章参考文献[11]~[13]。我们在这里只介绍一种基于 DFT 的 DCT 算法,以便理解 DCT 与 DFT 之间的关系。

令

$$W_K \equiv \exp(-\mathrm{j}2\pi/K)$$

则 K 个点的 DFT 可表示为

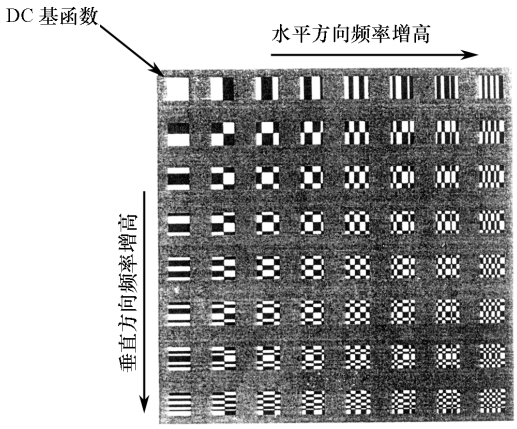
$$F(n) = \sum_{k=0}^{K-1} s(k) W_K^{nk} \qquad (n = 0,1,\cdots,K-1) \tag{3-71}$$

若有一个 N 点实数序列 $s(k)$,$k=0,1,\cdots,N-1$,我们定义一个与此序列相对于$(2N-1)/2$ 点对称的序列(见图 3-27),即

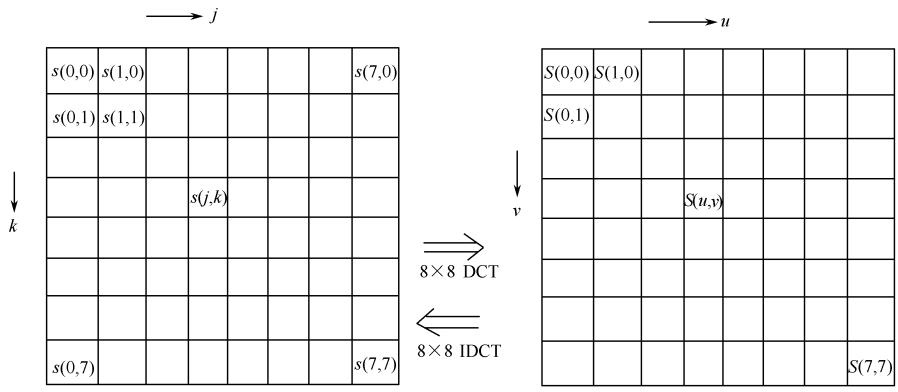
$$s(2N-k-1) = s(k) \qquad (k = N,N+1,\cdots,2N-1) \tag{3-72}$$

则整个 $K=2N$ 点序列的 DFT 可表示为

$$F(n) = \sum_{k=0}^{N-1} s(k) W_K^{nk} + \sum_{k=N}^{2N-1} s(2N-k-1) W_K^{nk} \tag{3-73}$$



(a) 二维 DCT 的基函数



(b)

图 3-26 (a)二维 DCT 的基函数 (b)$N=8$ 时的信号矩阵和变换系数矩阵

设 $i=2N-k-1$,并注意到 $W_K^{2N}=1$,上式则变为

$$F(n)=\sum_{k=0}^{N-1}s(k)W_K^{nk}+\sum_{i=0}^{N-1}s(i)W_K^{-n(i+1)} \tag{3-74}$$

用 k 代替 i,并在等式两边同乘以 $W_K^{n/2}/2$,则得到

$$\frac{1}{2}F(n)W_K^{n/2}=\sum_{k=0}^{N-1}s(k)\cos\frac{(2k+1)n\pi}{2N} \tag{3-75}$$

上式说明一个 N 点序列的 DCT 系数,可以由它对应的对称序列的前 N 个 $2N$ 点 DFT 系数乘以适当的数值得到。为了进一步简化上述关系,我们注意到,由于(3-75)式右端是实数,因此左端也应为实数。用 A_n 和 B_n 分别表示 $F(n)$ 的实数和虚数部分,则有

$$F(n)W_K^{n/2}=(A_n+\mathrm{j}B_n)\left(\cos\frac{\pi n}{2N}-\mathrm{j}\sin\frac{n\pi}{2N}\right) \tag{3-76}$$

将上式展开并令虚部为零,得到

$$B_n=A_n\tan\left(\frac{n\pi}{2N}\right) \tag{3-77}$$

将 B_n 代入(3-76)式得到

$$F(n)W_K^{n/2}=A_n\sec\left(\frac{n\pi}{2N}\right)=Re\{F(n)\}\sec\left(\frac{n\pi}{2N}\right) \tag{3-78}$$

式中 $Re\{F(n)\}$表示对函数 $F(n)$取实部。将上式代入(3-75)式,得到

$$\sum_{k=0}^{N-1} s(k)\cos\frac{(2k+1)n\pi}{2N} = \frac{1}{2}Re\{F(n)\}\sec\left(\frac{n\pi}{2N}\right) \tag{3-79}$$

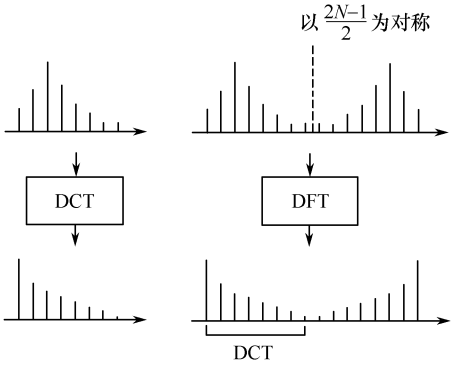


图 3-27　DCT 与 DFT 的关系

由上式可以得到如下结论：一个函数的 DCT 系数可以由该函数对应的偶函数的 DFT 系数的实部得到。图 3-27表示了这个过程。

二维的 DCT 可以直接计算，也可以通过这样的办法来实现：先按照行（或列）进行一维 DCT 变换，然后将变换结果再按列（或行）进行一维变换。

最后，有两点值得指出：

（1）我们知道，二维信号的傅氏变换的系数代表它所对应的空间频率分量的复振幅。(3-79)式表明，虽然 DCT 系数并不与空间频率分量的复振幅严格相等，但有一定的对应关系。特别是，$n=0$ 时的 DCT 系数与 DFT 的零频分量一样，代表空间域内信号的均值；

（2）如前所述，一个函数的 DCT 系数可以通过与该函数对应的偶函数的 DFT 系数得到。由于偶函数的对称性减小了 DFT 中由于周期延拓而产生的空间域中边缘的不连续性，从而使能量在频率域内更为集中。因此在数据压缩的应用中，DCT 比 DFT 具有更好的性能。

3.7　子带编码

3.7.1　子带编码的工作原理

子带编码(Subband Coding)，简称 **SBC**，与变换编码一样，是一种在频率域中进行数据压缩的方法。在子带编码中，如图 3-28 所示，首先用一组带通滤波器将输入信号分成若干个在不同频段上的子带信号，然后将这些子带信号经过频率搬移转变成基带信号，再对它们在奈氏频率上分别取样。取样后的信号经过量化、编码，并合成成一个总的码流传送给接收端。在接收端，首先把码流分成与原来的各子带信号相对应的子带码流，然后解码、将频谱搬移至原来的位置，最后经带通滤波、相加得到重建的信号。

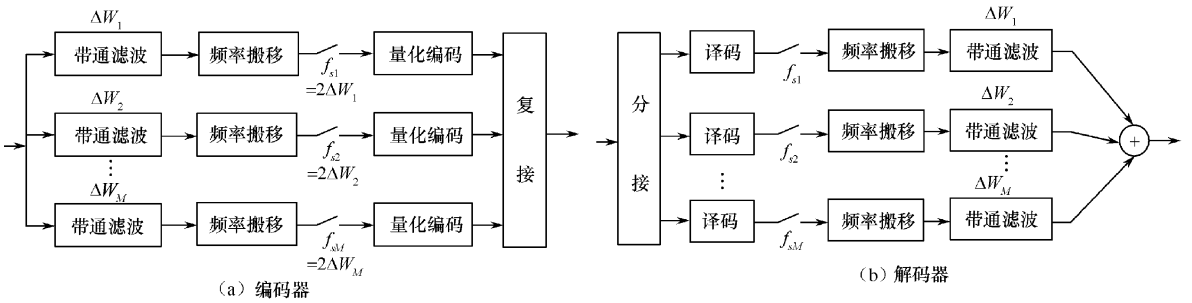


图 3-28　子带编、解码工作原理框图

在子带编码中，若各子带的带宽 ΔW_k 是相同的，则称为**等带宽子带编码**；若 ΔW_k 是互不相同的，则称为**变带宽子带编码**。

将信号分成若干个子带进行编码的好处是：

（1）可以利用人耳（或人眼）对不同频率信号的感知灵敏度不同的特性，在人的听觉（或视觉）不敏感的频段采用较粗糙的量化，从而达到数据压缩的目的；

(2) 各子带的量化噪声都束缚在本子带内，这就可以避免能量较小的频带内的信号被其他频段中的量化噪声所掩盖。

图 3-28 中所示的带通滤波器组是子带编码的重要部件，它影响到子带编码的复杂程度和性能。

首先，我们注意到，如果各子带的下截止频率 f_{lk} 恰好是该子带宽度 ΔW_k 的整数倍，即

$$f_{lk} = n\Delta W_k \qquad (k = 1,2,\cdots,M) \tag{3-80}$$

(其中 n 是一非负整数，M 为子带的个数)，则可以不必将带通信号搬移到基带上，直接以取样频率 $f_{sk}=2\Delta W_k$ 对子带信号取样，而不会产生混叠失真。以 $M=4$ 的变带宽子带为例，图 3-29给出了第三个子带的取样过程。图中(a)是输入信号的频谱；图(b)是第三个子带信号的频谱，其下截止频率 $f_{l3}=2\Delta W_3$；图(c)是以 $f_{s3}=2\Delta W_3$ 的频率对图(b)取样后，所得到的信号的频谱。

满足(3-80)式条件的子带滤波器组称为**整数子带滤波器组**。采用整数子带滤波器组省去了进行频谱搬移所需要的调制器和解调器，使系统得以简化。同时，我们还可以将带通滤波和取样的次序互换，即输入模拟信号先以 $f_s=2W$ 的频率取样，然后再经带通滤波和抽取获得各子带信号。其中 W 为输入信号的总带宽，按下式计算：

$$W = \sum_{k=1}^{M} \Delta W_k \tag{3-81}$$

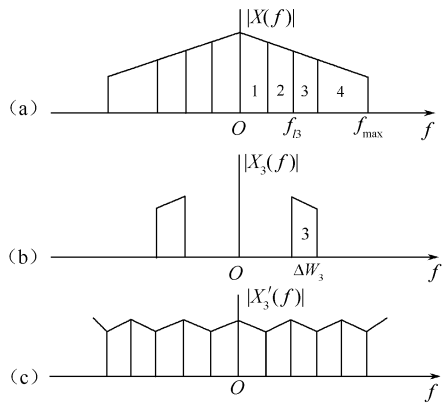


图 3-29　整数子带的取样过程

图 3-30(a)表示出使用抽取和内插的、$M=2$ 的子带编码和解码的原理方框图。此时，我们可以将系统的输入和输出看成是已在奈氏频率下取样的离散信号，图(b)给出了它对应的频谱。假设用带通滤波器取出图(b)中斜线所示的上子带，根据 3.4.1 节中讲述的原则，很容易得到经 2∶1 抽取后的上子带的频谱如图(c)所示。在解码端，再用 1∶2 的内插器，将频谱恢复到原来的位置。

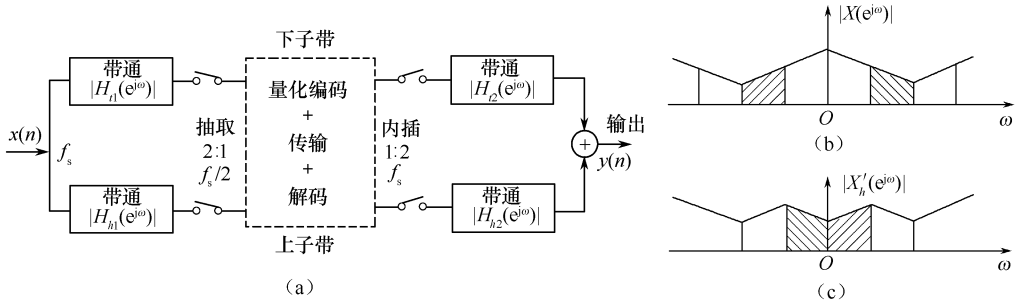


图 3-30　整数子带编码和解码原理

在采用整数子带滤波器的系统中，其总的传输速率 I 为

$$I = \sum_{k=1}^{M} f_{sk}R_k = \sum_{k=1}^{M} 2\Delta W_k R_k \tag{3-82}$$

式中，R_k 为第 k 个子带中每个样值量化的比特数。

对于子带编码的带通滤波器，我们注意到的第二个问题是，滤波器的具体实现不可能是理想的带通，其幅度响应不可避免地带有有限的滚降。因此在划分子带时，只能或者使子带间有交叠，如图 3-31(a)所示；或者使子带之间有一定的间隙，如图 3-31(b)所示。在图(a)的情况下，按照奈氏频率取样将会产生混叠失真；而在图(b)的情况下，由于原有信号的部分频带经滤波而损失掉了，重建的信号会有失真，例如对语音来讲，重建的语音会产生混响的主观感觉。3.7.2 节将要介绍的

正交镜像滤波器比较好地解决了混叠失真的问题，从而降低了对滤波器滚降特性的要求，简化了滤波器的复杂性，成为子带编码中最常采用的形式。

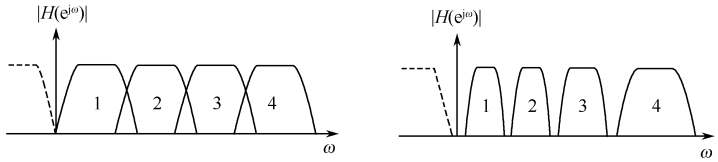


图 3-31　4 子带滤波器组的幅度响应

3.7.2　正交镜像滤波器组

图 3-32 给出了一个 $M=2$、等带宽的**正交镜像滤波器组**(Quadrature Mirror Filter Banks)的幅频特性图，其中上、下子带的幅频响应以 $f_s/2$ 为镜像对称，且有交叠。如果在图 3-30(a)中使用这样的滤波器，该图就是所对应的正交镜像滤波的子带编、解码系统。

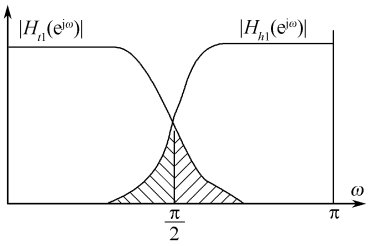


图 3-32　$M=2$ 的正交镜像滤波器组的幅频特性

由于系统的输入是在奈氏频率上对信号取样所得到的离散序列，因此在 2∶1 抽取后，下子带的大于 $f_s/2$ 的部分会以 $f_s/2$ 为轴折叠到上子带中，而上子带的小于 $f_s/2$ 的部分会以 $f_s/2$ 为轴折叠到下子带中(见图 3-32 中斜线部分)。在一般的情况下这种混叠失真是不能消除的，但是如果满足下面所介绍的条件，失真在子带解码过程中将得以抵消。对于具有偶数个抽头(即 N 为偶数)的对称及反对称的 FIR 滤波器：

$$h_l(n) = h_l(N-1-n) \qquad (n=0,1,\cdots,N/2-1) \tag{3-83}$$

$$h_u(n) = -h_u(N-1-n) \qquad (n=0,1,\cdots,N/2-1) \tag{3-84}$$

这些条件是

$$h_u(n) = (-1)^n h_l(n) \qquad (n=0,1,\cdots,N-1) \tag{3-85}$$

$$h'_l(n) = h_l(n) \qquad (n=0,1,\cdots,N-1) \tag{3-86}$$

$$h'_u(n) = -h_u(n) \qquad (n=0,1,\cdots,N-1) \tag{3-87}$$

而且

$$|H_l(e^{j\omega})|^2 + |H_u(e^{j\omega})|^2 = 1 \tag{3-88}$$

其中 $h_l(n)$ 和 $h_u(n)$ 分别为编码器中的下子带和上子带滤波器的冲激响应，$H_l(e^{j\omega})$ 和 $H_u(e^{j\omega})$ 分别为 $h_l(n)$ 和 $h_u(n)$ 的傅氏变换，而 $h'_l(n)$ 和 $h'_u(n)$ 则分别为解码器中的下、上子带滤波器的冲激响应。

(3-88)式表示的条件代表频域的全通特性。除了 $N=2$ 和 N 接近于无穷大的情况以外，满足(3-85)式条件的镜像滤波器不能满足(3-88)式的要求。但是，通过谨慎地选择 N 的大小，可以极其接近(3-88)式的要求。参考文献[14,15]中给出了 N 在 8～64 范围内符合上述条件的正交镜像滤波器的系数值。

下面我们证明，在不考虑量化失真和信道误码的情况下，满足(3-83)式至(3-88)式条件的正交镜像滤波器组能够消除混叠失真，正确地重建输入信号。设输入序列 $x(n)$ 的 Z 变换为 $X(z)$，经过下子带滤波器后，得到

$$X_l(z) = X(z) \cdot H_l(z) \tag{3-89}$$

根据数字信号处理的有关理论(见本章习题 3)，经 2∶1 抽取后的下子带信号 $l(n)$ 的 Z 变换 $L(z)$ 为

$$L(z) = \frac{1}{2}[X_l(z^{1/2}) + X_l(-z^{1/2})] \tag{3-90}$$

经 1∶2 内插之后为

$$L'_l(z) = L(z^2) \tag{3-91}$$

再经 $H'_l(z)$(即 $h'_l(n)$ 的 Z 变换)的滤波,得到

$$X'_l(z) = L'_l(z)H'_l(z) \tag{3-92}$$

将(3-89)式至(3-91)式代入(3-92)式,得到

$$X'_l(z) = \frac{1}{2}[X(z)H_l(z) + X(-z)H_l(-z)]H'_l(z) \tag{3-93}$$

用相类似的方法,可以得到重建的上边带信号 $x'_u(n)$ 的 Z 变换 $X'_u(z)$ 为

$$X'_u(z) = \frac{1}{2}[X(z)H_u(z) + X(-z)H_u(-z)]H'_u(z) \tag{3-94}$$

因此,输出信号 $x'(n)$ 的 Z 变换 $X'(z)$ 为

$$\begin{aligned} X'(z) &= X'_l(z) + X'_u(z) \\ &= \frac{1}{2}[H_l(z)H'_l(z) + H_u(z)H'_u(z)]X(z) + \\ &\quad \frac{1}{2}[H_l(-z)H'_l(z) + H_u(-z)H'_u(z)]X(-z) \end{aligned} \tag{3-95}$$

上式等号右边的第一项代表所需要的重建信号,第二项代表混叠失真。由(3-85)式至(3-87)式,我们有

$$H_u(z) = H_l(-z) \tag{3-96}$$

$$H'_l(z) = H_l(z) \tag{3-97}$$

$$H'_u(z) = -H_u(z) \tag{3-98}$$

将(3-96)式至(3-98)式代入(3-95)式,得到

$$X'(z) = \frac{1}{2}[H_l^2(z) - H_u^2(z)]X(z) \tag{3-99}$$

从上式可以看出,(3-85)式至(3-87)式所表示的条件保证了使(3-95)式中的混叠失真项为零。由(3-83)式和(3-84)式所示的子带滤波器的对称性,我们可以导出

$$H_l(e^{j\omega}) = |H_l(e^{j\omega})|e^{-j\omega(N-1)/2} \tag{3-100}$$

$$H_u(e^{j\omega}) = |H_u(e^{j\omega})|e^{-j\omega(N-1)/2} \cdot e^{j\pi/2} \tag{3-101}$$

将(3-100)式和(3-101)式代入(3-99)式,并考虑到 $j = e^{j\omega}$ 和(3-88)式的条件,得到

$$X'(e^{j\omega}) = \frac{1}{2}X(e^{j\omega})e^{-j\omega(N-1)} \tag{3-102}$$

上式说明正交镜像子带编、解码可以准确地恢复出原有的信号,而仅引入($N-1$)个取样间隔的延时。

3.7.3 时域混叠消除

时域混叠消除法(Time Domain Alias Cancellation,**TDAC**)也是一种子带编码,它与正交镜像滤波方法之间具有对偶性。正交镜像滤波在频率域内抵消混叠失真,TDAC 则在时间域内抵消混叠失真。

假设有图 3-33(a)所示的系统,其输入信号为一个长时间序列 $y(n)$(如宽带的数字音频信号),用一个长度为 K 的窗 $h(n)$ 将其截断,将截取的部分 $y(n)h(n)$ 变换到频率域得到 K 个变换系数 $X(k)$,$k=0,1,K-1$。然后将窗移动 $K/2$,重复上述过程。这样,就把一个时间序列分解成了 K 个变换系数序列[见图 3-33(b)],这一过程称为**分析**。

由变换系数还原为时间序列的过程称为**综合**。首先将一组变换系数(即变换系数序列在同一时间 m 的样值)经反变换后得到 K 个时间域的样值 $g(n)$。然后用一个同样长度的窗 $f(n)$ 对 $g(n)$ 加权。依次对每组变换系数都进行上述处理,并将所得到的具有 K 个样值的时域序列逐次移动 $K/2$ 个样点后相加,就还原出一个长时间序列 $\hat{y}(n)$。

在图 3-33(a)所示系统中使用的变换不是一般的 DFT 或 DCT,而是**修正的(Modified)**离散余

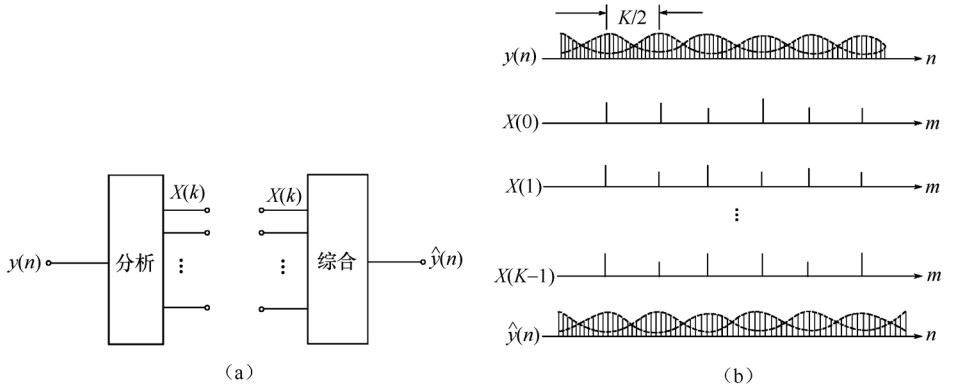


图 3-33　MDCT/MDST 分析与综合系统

弦和离散正弦变换，简称 **MDCT/MDST**。MDCT 的正变换与反变换分别定义如下：

$$X(k)=\frac{1}{K}\sum_{n=0}^{K-1}x(n)\cos\left[\frac{2\pi k}{K}(n+n_0)\right]\quad(k=0,1,\cdots,K-1)\tag{3-103}$$

$$x(n)=\frac{1}{K}\sum_{k=0}^{K-1}X(k)\cos\left[\frac{2\pi k}{K}(n+n_0)\right]\quad(n=0,1,\cdots,K-1)\tag{3-104}$$

其中 $n_0=(K/2+1)/2$ 为一固定的时间偏移。与 MDCT 类似，MDST 的正、反变换分别定义为

$$X(k)=\frac{1}{K}\sum_{n=0}^{K-1}x(n)\sin\left[\frac{2\pi k}{K}(n+n_0)\right]\quad(k=0,1,\cdots,K-1)\tag{3-105}$$

$$x(n)=\frac{1}{K}\sum_{k=0}^{K-1}X(k)\sin\left[\frac{2\pi k}{K}(n+n_0)\right]\quad(n=0,1,\cdots,K-1)\tag{3-106}$$

系统在分析过程中交替地使用 MDCT 和 MDST。将窗函数从初始位置开始移动的次数记为 m。当 m 为偶数时，用 MDCT 对窗所截取的数据进行变换，m 为奇数时使用 MDST 变换。从(3-103)式和(3-105)式，我们注意到

$$X(k)=X(K-k)\qquad(k=0,1,\cdots,K/2)\qquad(\text{MDCT})\tag{3-107}$$

$$X(k)=-X(K-k)\qquad(k=0,1,\cdots,K/2)\qquad(\text{MDST})\tag{3-108}$$

即 K 个点的 MDCT/MDST 只产生 $\frac{K}{2}+1$ 个独立的系数。

如果我们将分析所得到的变换系数序列(仅指独立的变换系数)经时分复用后构成一个输出序列，并要求输出序列与输入序列 $y(n)$ 具有相同的比特率，那么每个变换系数序列的比特率只能为输入比特率的 $2/K$。满足这一要求的系统称为**临界取样系统**。从图 3-33(b)可以看出，窗 $h(n)$ 每次移动 $K/2$，正好将 $X(k)$ 的速率降低到临界取样所要求的数值。

现在来看信号的综合过程。由于 MDCT 和 MDST 正变换所产生的独立系数的个数少于 K，因此反变换不可能精确地恢复出原来的时间序列。以 MDCT 为例，将(3-103)式代入(3-104)式，并经简单的三角关系的推导，得到

$$\hat{x}(n)=\frac{1}{2K}\sum_{r=0}^{K-1}x(r)\left\{\sum_{k=0}^{K-1}\cos\left[\frac{2\pi k}{K}(r-n)\right]+\sum_{k=0}^{K-1}\cos\left[\frac{2\pi k}{K}(r+n+2n_0)\right]\right\}\tag{3-109}$$

考虑到

$$\sum_{k=0}^{K-1}\cos\left(\frac{2\pi k}{K}n\right)=\begin{cases}K & (n=sK,\ s\text{ 为整数})\\ 0 & (\text{其他})\end{cases}\tag{3-110}$$

得到

$$\hat{x}(n)=\frac{x(n)}{2}+\frac{x(K-n-2n_0)}{2}\tag{3-111}$$

由上式看出，反变换的结果是在原时间序列上再加上一个混叠项。由于离散变换周期性延拓

的特点，混叠项中的负时间值可看作是高端的折叠，例如，$x(-1)=x(K-1)$。

与 MDCT 类似地，可以得到 MDST 反变换的结果为

$$\hat{x}(n)=\frac{x(n)}{2}-\frac{x(K-n-2n_0)}{2} \tag{3-112}$$

在图 3-33(a)所示的系统中交替地使用 MDCT/MDST，其综合过程的输出正好使得在时间域内的混叠相互抵消，精确地恢复出原序列。我们现在以 $m=i$(i 为偶数)的一段数据为例说明这一过程。在输出序列 $\hat{y}(n)$ 中，$n=0,1,\cdots,K/2-1$ 的值是本段($m=i$) 数据经 MDCT 分析/综合的结果和前一段($m=i-1$) 数据经 MDST 分析/综合的结果之和，即

$$\begin{aligned}\hat{y}_i(n)=f_i(n)\left[\frac{y_i(n)h_i(n)}{2}+\frac{y_i(K-n-2n_0)\,h_i(K-n-2n_0)}{2}\right]+f_{i-1}\left(n+\frac{K}{2}\right)\\ \left[\frac{y_{i-1}\left(n+\frac{K}{2}\right)h_{i-1}\left(n+\frac{K}{2}\right)}{2}-\frac{y_{i-1}\left(\frac{K}{2}-n-2n_0\right)h_{i-1}\left(\frac{K}{2}-n-2n_0\right)}{2}\right]\\ \left(n=0,1,\cdots,\frac{K}{2}-1\right)\end{aligned} \tag{3-113}$$

其中 $h(n)$ 和 $f(n)$ 分别为分析窗和综合窗函数。假设 $f(n)=h(n)$ 和 $h(K-1-n)=h(n)$。考虑到 $m=i$ 和 $m=i-1$ 的两个窗有 50%的重叠，即

$$y_i(n)=y_{i-1}\left(n+\frac{K}{2}\right) \tag{3-114}$$

则(3-113)式可变为

$$\hat{y}_i(n)=\frac{y_i(n)\left[h_i^2(n)+h_i^2\left(n+\frac{K}{2}\right)\right]}{2}\qquad(n=0,1,\cdots,K/2-1) \tag{3-115}$$

因此，只要满足下式：

$$h_i^2(n)+h_i^2\left(n+\frac{K}{2}\right)=2 \tag{3-116}$$

则 $\hat{y}(n)=y_i(n)$，$(n=0,1,\cdots,K/2-1)$，即可精确地恢复出原来的时间序列。图 3-34 描述了上述分析/综合的过程。

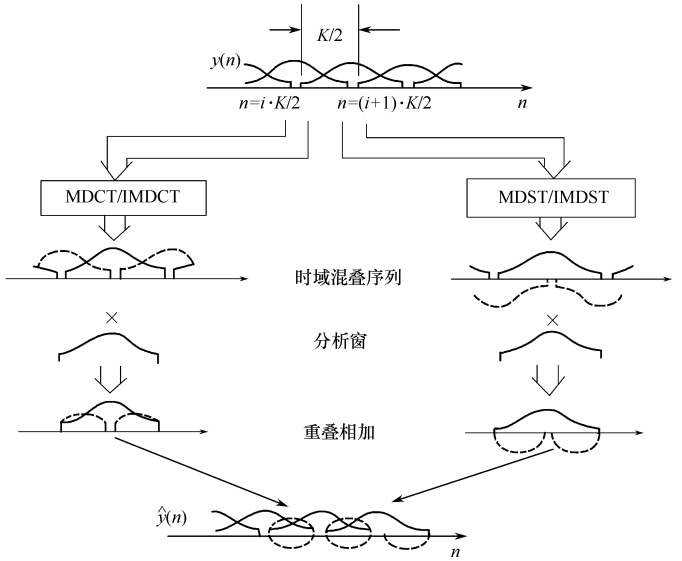


图 3-34　TDAC 编码中混叠失真的抵消

3.8 小波变换编码

小波变换与正交变换类似，是对信号进行分析的一种工具。我们知道，信号经傅氏变换（或余弦变换）后被分解成许多不同频率的分量，即分析的结果在频域内有精细的分辨率；但是从理论上讲正弦波在时域（空间域）上是无限伸展的函数，因此傅氏分析在时域上不具有分辨能力，它只能刻画信号在整个时间域（$-\infty,+\infty$）上的频谱特性，而不能反映信号在时间的局部区域上的频率特征。与傅氏分析不同，小波分析在时域和频域上同时具有良好的局部化性质，可以通过伸缩和平移对信号进行多尺度分析，解决傅氏变换等无法解决的许多问题。小波变换也因此被誉为“数学显微镜”。

3.8.1 多尺度分析

设函数 $\Psi\in L^2(\mathrm{R})$，R 为实数集，若 Ψ 满足允许条件：

$$C_\Psi = 2\pi\int \frac{|\hat{\Psi}(\omega)|^2}{|\omega|}\mathrm{d}\omega < \infty \tag{3-117}$$

其中 $\hat{\Psi}(\omega)$ 为 $\Psi(x)$ 的傅氏变换，则称 Ψ 为**基小波**，或**母小波**。由允许条件可知 Ψ 满足 $\hat{\Psi}(0)=0$ 和 $\hat{\Psi}(\infty)=0$，因此 Ψ 实质上为一带通滤波器。

相应于 Ψ 的小波变换为

$$W_\Psi(f,a,b) = |a|^{-\frac{1}{2}}\int_{-\infty}^{+\infty} f(x)\Psi\left(\frac{x-b}{a}\right)\mathrm{d}x \tag{3-118}$$

而

$$\Psi_{a,b}(x) = |a|^{-\frac{1}{2}}\Psi\left(\frac{x-b}{a}\right) \tag{3-119}$$

称为**小波**，不难看出它是母小波经平移和伸缩后的结果，其中 a 和 b 分别为伸缩及平移因子。若将 a,b 离散化（$a=a_0^m, b=nb_0a_0^m$），就可以得到离散的小波函数，例如取 $a_0=2, b_0=1$ 时，就得到一个**二进小波**，$\Psi_{m,n}(x)=2^{-\frac{m}{2}}\Psi(2^{-m}x-n)$，$m,n\in Z$，$Z$ 为整数集。二进小波变换是数字图像处理中最常用的一类小波。

由(3-118)式可得，二进小波变换的变换系数为

$$C_{m,n}(f) = 2^{-m/2}\int_{-\infty}^{+\infty} f(x)\Psi(2^{-m}x-n)\mathrm{d}x \tag{3-120}$$

由小波系数重构的原始信号则为

$$f(x) = \sum_{m=-\infty}^{+\infty}\sum_{n=-\infty}^{+\infty} C_{m,n}(f)\Psi_{m,n}(x) \tag{3-121}$$

(3-120)式所示的变换称为小波级数，其中小波系数为离散的，但信号 $f(x)$ 仍为连续函数。当输入信号 $f(x)$ 和小波伸缩及平移因子（a 和 b）均为离散时，则变换称为**离散小波变换** DWT（Discrete Wavelet Transform）。

设平方可积函数空间 $L^2(R)$ 上的闭子空间序列$\{V_m\}$，$m\in Z$，若满足：

① $\cdots\subset V_1\subset V_0\subset V_{-1}\subset\cdots$，即每一个子空间都包含在下一个尺度（分辨率增加一倍）的子空间内；

② $\overline{(\bigcup_{m\in Z}V_k)}=L^2(R)$，即子空间的并集在 $L^2(R)$ 中是稠密的；

③ $(\bigcap_{m\in Z}V_m)=0$，即 $L^2(R)$ 中的每一个非零 $f(x)$ 都具有非零尺度，当 $m\to\infty$ 时，它的投影 $f^{(m)}(x)$ 在 $L^2(R)$ 意义上收敛于 0；

④ $f(x)\in V_m\Leftrightarrow f(2x)\in V_{m-1}$，$m\in Z$，即子空间 V_m 中的函数尺度增加一倍，则成为较高尺度 V_{m-1} 内的函数；

⑤ $f(x)\in V_m \Leftrightarrow f(x-2^m n)\in V_m, m,n\in Z$，即 V_m 中的函数平移后仍在此子空间内(分辨率不变)；

⑥ 存在函数 $\Phi(x)\in L^2(R)$，使得$\{\Phi_{0,n}(x): n\in Z\}$构成 V_0 的一组标准正交基，其中$\Phi_{m,n}(x)=2^{-\frac{m}{2}}\Phi(2^{-m}x-n), m,n\in Z$；

则$\{V_k\}$形成一个**二进多尺度分析**，$\Phi(x)$称为**尺度函数**(Scaling function)。

由④、⑤、⑥可知，固定 m，$\{\Phi_{m,n}(x): n\in Z\}$构成了 V_m 的一组正交基。$\Phi(x)$的存在性也是显而易见的，例如取 $\Phi(x)$区间$[0,1]$上的示性函数，即当 $x\in[0,1]$时，$\Phi(x)=1$，其他情况$\Phi(x)=0$)，就构成了 V_0 的一组标准正交基。

当一组闭子空间序列满足上述(1)～(6)时，可以证明存在 $L^2(R)$的一组小波正交基$\{\Psi_{m,n}(x): m, n\in Z\}$，使得$\{\Psi_{m,n}(x): n\in Z\}$为 V_{m-1}中关于 V_m的正交补空间 W_m 的一组标准正交基，即

$$V_{m-1} = V_m \oplus W_m, m \in Z \tag{3-122}$$

既然 m 是任意整数，因此有

$$V_{m-1} = V_{m+1} \oplus W_{m+1} \oplus W_m \tag{3-123}$$

我们还可以以此类推，因此由多尺度分析可得：$L^2(R)=\oplus_{m\in Z}W_m$ 或 $L^2(R)=V_J\oplus_{m=-\infty}^{J}W_m$。

上面的式子说明，某一子空间中的任意函数(信号)可以分解为在较低尺度子空间的该信号的一个近似值，再加上一系列由小波分解产生的函数，后者可以认为是信号在 W_m 上的投影，代表了信号取近似值后损失的细节信息。

可以证明[16]，尺度函数与小波函数之间存在双尺度关系：

$$\Phi(x) = \sum_n h_n \Phi(2x-n) \tag{3-124}$$

$$\Psi(x) = \sum_n g_n \Phi(2x-n) \tag{3-125}$$

$\{\Psi_{m,n}(x): m,n=\in Z\}$构成 $L^2(R)$的标准正交小波基，也就是说，通过多尺度分析的尺度函数 $\Phi(x)$可以构造标准正交母小波 $\Psi(x)$。(3-124)式和(3-125)式是下一节介绍的级联算法的一个基础，其中$\{h_n\}$和$\{g_n\}$是对应的数字低通滤波器和高通滤波器的冲激响应。

3.8.2 二进小波变换

1. 二进小波分解

设$\{c_{0,n}\}$，$L_0\leqslant n\leqslant L_1$ 为初始信号，$\{h_n\}$和$\{g_n\}$分别为二进小波分解低通滤波器和高通滤波器，也称为**分析滤波器**，则有小波分解公式：

$$c_{m,n} = \sum_j h_{j-2n} c_{m-1,j} \tag{3-126}$$

$$d_{m,n} = \sum_j g_{j-2n} c_{m-1,j} \tag{3-127}$$

其中$\{c_{m,n}\}$为第 m 次小波分解后所得到的低通信号，即在 V_m 上信号的近似值；$\{d_{m,n}\}$为相应的高通信号，即在 W_m 上的细节信息。由于二进小波分解的每一个尺度分辨率降低一倍，所以两个输出信号分别以简单的“2 抽 1”方式进行下取样，然后对于在 m 尺度上的信号近似值 $c_{m,n}$进行 $m+1$ 尺度上的继续分解……图 3-35 的左侧给出了一个三层的级联分解过程。

2. 二进小波重构

设$\{\tilde{h}_n\}$和$\{\tilde{g}_n\}$分别为二进小波重构低通滤波器和高通滤波器，也称为**综合滤波器**，m 次小波分解后的信号如(3-126)式和(3-127)式，则有小波重构公式：

$$c_{m-1,i} = \sum_j \tilde{h}_{i-2j} c_{m,j} + \sum_j \tilde{g}_{i-2j} d_{m,j} \tag{3-128}$$

图 3-35 右侧给出了信号的重构过程。

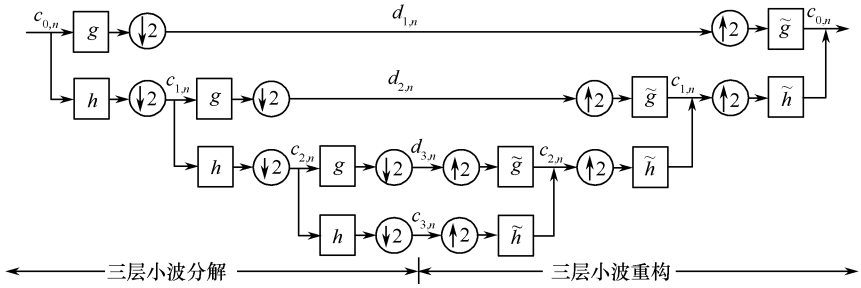


图 3-35　三层小波分解和重构

对于正交小波，重构滤波器与分解滤波器是相同的。但是，由于对称的正交小波只有简单的 Haar 小波，而对称小波滤波器在数字信号处理中具有保持线性相位和降低小波变换复杂度等优势，所以在进行数字滤波时往往采用**对称双正交小波滤波器**，常用的有 Daubechies 5-3 滤波器组和 9-7 滤波器组。

3. 二维信号的小波分解

对于二维图像信号的小波分解，通常先对二维信号按行做一维小波分解，然后再按列做一维小波分解。重构时，依次进行相应的列信号重构与行信号重构即可恢复原信号。

图 3-36(a)表示出二维小波分解过程。行方向分解生成左边为低通子带、右边为高通子带的中间结果；再经列方向分解后形成 LL、LH、HL 和 HH 4 个子带，其中 LL 是原始图像的粗略近似，LH、HL 和 HH 分别包含水平、垂直和 45°倾斜的高频分量。图(b)给出两层分解的子带分布情况；图(c)则给出一个实际图像经过 2 层小波分解后的结果。由图(c)看出，经小波变换后能量多集中在 LL 子带内。

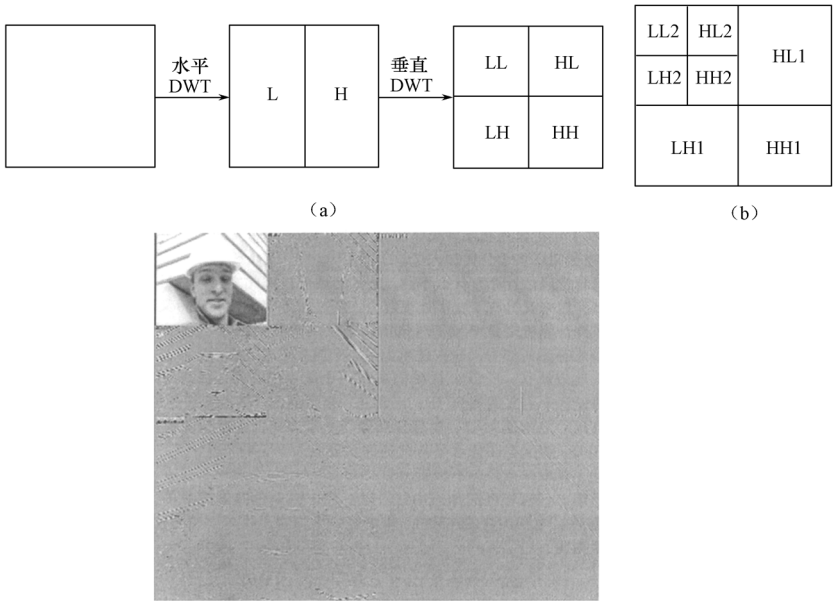


图 3-36　二维信号的小波分解

3.8.3　变换系数的排序和编码

经小波变换后信号能量已经集中在少数系数上，将幅值很小的其他系数忽略(量化到 0)，则可达到数据压缩的目的。不过在利用小波变换进行图像编码时一个值得注意的问题是，量化后的变

换系数矩阵是一个具有少量非零值和大量零值的稀疏矩阵，在将此矩阵转换成一维序列时，如何有效地组织非零系数和有效地表达零系数所在的位置，对数据压缩的效率有重要的影响。本节讨论与此有关的问题。

1. 嵌入零树小波编码

人们发现，如果低频子带中某个系数为非零值，则在高频子带对应位置上的系数也为非零值的概率很大。根据这个特点，Lewis 和 Knowles 在 1992 年提出了**小波零树编码**算法。在小波零树编码算法中，图像经过 N 层离散二维小波变换后得到小波系数，这些系数按照子带的相同方向形成小波树结构。图 3-37 为图像经过三层离散二维小波变换后形成的小波树结构示意图。其中：

① 低频子带 LL3 的每一个小波系数为树的根节点，它在同层小波变换的高频子带 HL3、LH3 和 HH3 中相同空间位置上均有一个子节点；

② 每个高频子带的小波系数在同方向较高频子带上具有 4 个子节点；

③ 最高频子带的每个小波系数没有子节点，因此也被称为叶节点；

④ 父节点是与子节点相对的一个称呼，B 是 A 的子节点，则称 A 为 B 的父节点；

⑤ 一个节点的子节点及其子节点的子节点等称为该节点的后代；

⑥ 一个节点的父节点及父节点的父节点等称为该节点的祖先；

⑦ 小波系数与它的后代组成一个小波树。

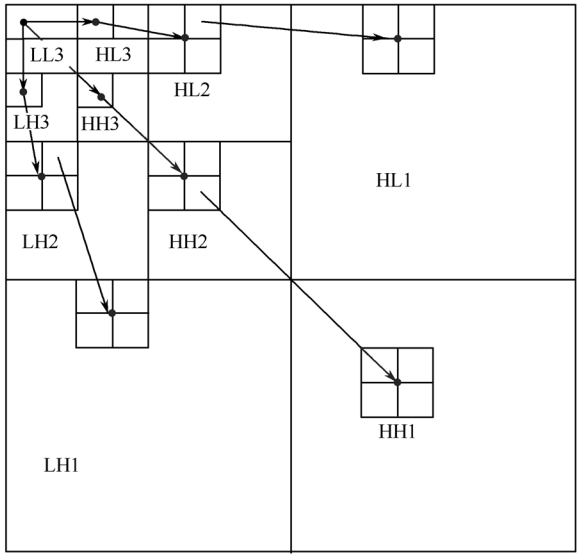


图 3-37　EZW 编码的小波树结构示意图

小波零树编码就是利用小波树的强相关性，将父节点的绝对值与门限值进行比较，当父节点绝对值小于门限值时，认为该小波树均不是显著系数，因此将该小波树都以零值编码，形成大量的小波零树，从而减少数据。零树编码不是十分完美，譬如父节点不显著时，也存在后代节点是重要的情形，此时忽略子孙节点将造成较大的误差。

1993 年 Shapiro 在小波零树编码算法的基础上提出了**嵌入零树小波编码**(Embedded Zerotree Wavelet，**EZW**)算法[17]。在 EZW 算法中，用 5 种符号表示小波系数：正符号、负符号、孤立零值、小波零树和零符号。正符号(Positive Symbol，POS)表示绝对值大于门限值，且值为正的小波系数；负符号(Negative Symbol，NEG)表示绝对值大于门限值，且值为负的小波系数；孤立零值(Isolated Zero，IZ)表示绝对值小于门限值，但是存在后代节点的绝对值大于门限值的小波系数；小波零树(Zerotree，ZTR)表示绝对值小于门限值，且后代节点的绝对值都小于门限值的小波系数；零符号(Zero Symbol，

Z)表示绝对值小于门限值的叶节点。由于在规定的扫描顺序下小波树的位置能完全确定下来，所以在实际编码中小波零树和零符号可以用同一符号来表示。

(1) 门限值的选取

初始门限值：$T_0=2^{\lfloor \log_2(max|V_{DWT}|)\rfloor}$，其中 V_{DWT} 表示小波系数，$\lfloor\cdot\rfloor$ 表示取不超过该数的最大整数值。此后每次扫描的门限值：$T_n=\frac{T_{n-1}}{2}$，$n=1,2,\cdots$。

(2) 扫描顺序

在编码过程中，对小波系数通常采用如图 3-38 所示的 Zig-Zag(Z)顺序进行多次扫描，前一次扫描获得的数据比后一次更重要。

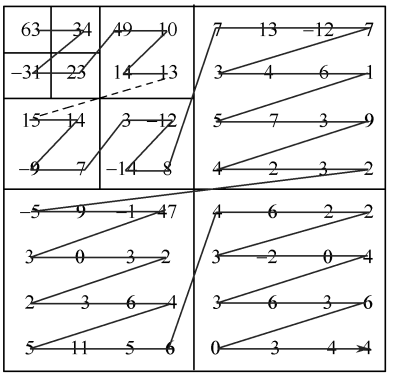


图 3-38　对 3 层分解的小波系的扫描

(3) 编码过程

编码主要分为三个过程：主扫描过程(dominant pass)，主要确定小波系数的符号；辅扫描过程(refinement pass)，对小波显著系数(正符号和负符号)进行加细量化；符号编码过程(symbol encode)，对符号进行熵编码。

首先，在第 i 次主扫描过程中，根据门限 T_{i-1} 确定每个系数对应的符号。绝对值大于门限值的系数，称为显著系数，依据它的正、负赋予符号 POS 或 NEG。绝对值小于门限值的系数，如果其子孙节点系数的绝对值均小于门限值，则它们构成零树，赋予该节点 ZTR，并不再对其子孙节点进行扫描；如果其子孙节点中有一个系数的绝对值大于门限值，则赋予该节点 IZ，对其子孙节点的扫描照常进行。将扫描得到的符号按顺序记录在显著系数表中。

第 i 次辅助扫描是对当前的显著系数表进行扫描，以对符号为 POS 和 NEG 的系数的绝对值进行加细量化。将$[T_{i-1},2T_0)$划分成$[T_{i-1},2T_{i-1})$大小的若干个区间，若系数的绝对值落入某一个区间的下半部(小于区间中点值)，输出“0”；否则，输出“1”。将输出的“0”和“1”按扫描顺序记录在加细量化表中。

然后，在新门限值 T_i 下进行第 $i+1$ 次主扫描和辅扫描。在这次主扫描中，前面已经确定为显著系数的节点将不再扫描，同时将该节点的系数看成 0，以确保更多的零树产生。主扫描和辅扫描得到的结果将分别存入显著系数表和加细量化表中(附加在第 i 次扫描结果之后)。

上述的两步扫描过程继续进行，直至门限值为 1。最后，对所得的两个表进行熵编码得到 EZW 码流。编码中的符号 IZ 和 Z 可以合并为 Z。

由上看出，EZW 编码对数据进行逐次逼近量化，将重要性高的数据优先编码(在表的前面)，而且可以在任意一点停止扫描编码，形成重建质量不同的码流，保留码流越长，系数量化精度越细，重建图像质量越高。这种编码方式称为**嵌入式编码**。多尺度分析的特性使小波变换编码具备良好的嵌入式编码性能。

2. 其他高性能的小波编码算法

Said 和 Pearlman 受 EZW 算法的启发，在 1996 年提出了**多级树集合分裂算法**(Set Partitioning In Hierarchical Trees，**SPIHT**)[18]。SPIHT 算法依然利用小波零树特性进行编码，尽管与 EZW 算法有很多相似之处，但也存在一些明显的区别。第一，SPIHT 算法在小波树结构上与 EZW 算法有所不同，同时定义了两种零树，比 EZW 划分的更为细致；第二，EZW 编码的扫描顺序是一种固定的光栅扫描，而 SPIHT 编码的扫描顺序取决于以前编码的树的显著性；第三，显著性扫描过程与加细量化过程与 EZW 算法的顺序相反。与 EZW 算法一样，SPIHT 算法也是一种嵌入式编码方法，它可以在任意点截断比特流，得到不同质量的图像数据。

EZW 算法和 SPIHT 算法都利用了小波变换中存在的父－子节点的相关性。然而更进一步的研究表明，基于小波变换的图像编码算法编码增益更多的得益于小波变换的能量压缩性质，而不是特定的图像数据的相关性。因此，对每个子带进行独立的编码也可以获得良好的编码性能。

Taubman 在 2000 年提出的**基于优化截断嵌入式块编码**(Embedded Block Coding with Optimal Truncation，**EBCOT**)[19]就是一种对各子带进行独立编码的方法。EBCOT 算法将每个子带分成相对较小的块(64×64 或 32×32)，这些块称为“码块(code-block)”，每个码块独立进行编码，产生基本的嵌入式流。最后由这些基本嵌入式比特流可以合成一个总嵌入式流，称为“组合流(packet-stream)”。合成组合流时，每个码块截断点比特流长度与失真的关系是已知的，可以根据这种关系优化截断各码块比特流以获得效率更高的组合流，这种优化截断方法称为压缩后率失真优化(Post-Compression Rate-Distortion optimization，PCRD-opt)方法。EBCOT 算法是对每个码块进行独立编码的，因此通过 EBCOT 算法不仅可以实现失真的可伸缩性，通过选择不同子带的码块比特流还可以实现分辨率的可伸缩性，结构比 EZW 和 SPIHT 算法要更灵活。同时对每个码块进行独立编码也为在硬件中实现并行算法提供了可能性。当压缩比特流出现误码时，误码的影响也仅出现在相应的码块中，并不影响其他码块的解码，因此 EBCOT 算法还具有良好的抗误码性能，适合在网络中进行传输。EBCOT 算法的这些特性促成了 JPEG2000 采用 EBCOT 编码方法[19]。

3.9 量化

量化是将具有连续幅度值的输入信号转换到只具有有限个幅度值的输出信号的过程。量化后的信号再经反量化一般不能精确地恢复原有值，即存在信息缺失。在压缩编码中，量化是一个对压缩后的码率和重建图像质量均产生重要影响的步骤。

3.9.1 均匀量化器

在量化器中从输入信号 x 到输出信号 y 的转换过程可以表示为

$$y = Q(x) = y_i \qquad (x \in A_i) \tag{3-129}$$

$$A_i : \{x_i < x \leqslant x_{i+1}\}$$

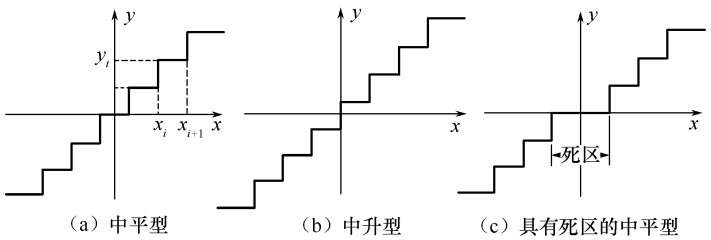
式中，x_i 为判决电平；y_i 为输出电平。

图 3-39 给出了几种均匀量化器的量化特性 $Q(x)$。图中量化器的特性都是对称的，且

$$y_{i+1} - y_i = \Delta \tag{3-130}$$

$$x_{i+1} - x_i = \Delta \tag{3-131}$$

式中，Δ 称为**量化台阶**或**量化步长**。图 3-39(a)所示的量化特性称为**中平型**(Mid-Step，或 Mid-Tread)，而图(b)所示的称为**中升型**(Mid-Riser)。二者的区别仅在于输出电平是否包括了零电平。除了(a)、(b)两种特性以外，在压缩编码中还常常用到图(c)所示的量化特性。图(c)与(a)所示特性的区别在于输出零电平所对应的输入电平范围较宽，因而有利于数据的压缩。这个较宽的范围通常称为**死区**(Dead zone)。



(a) 中平型　(b) 中升型　(c) 具有死区的中平型

图 3-39　均匀量化器的量化特性

量化器输出幅度与输入幅度之差，称为**量化误差**，其均方误差值 σ_d^2 为

$$\sigma_d^2 = <[x-Q(x)]^2> = \sum_{i=1}^{N}\int_{x_i}^{x_{i+1}}(x-y_i)^2 p(x)\mathrm{d}x \tag{3-132}$$

式中，$p(x)$ 为量化器输入信号 x 的概率分布密度，N 为量化级数。

3.9.2 最小均方误差量化器

如果输入信号的统计特性[如概率密度函数 $p(x)$]是已知的，则可以通过使均方量化误差 σ_d^2 达到最小，来设计量化级数为 N 的量化器的输出电平 y_i 和判决电平 x_i。这样得到的量化器通常称为**劳依得—麦克思**(Loyd-Max)**量化器**。

使均方量化误差

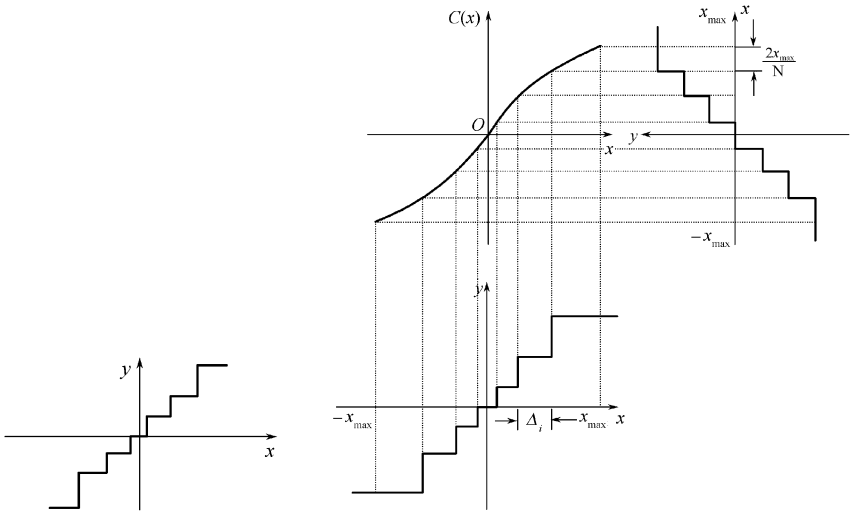
$$\sigma_d^2 = \sum_{i=1}^{N}\int_{x_i}^{x_{i+1}}(x-y_i)^2 p(x)\mathrm{d}x \tag{3-133}$$

为最小的必要条件是，σ_d^2 对 x_i 和 y_i 的偏导数均为零。由此得到

$$x_{i,\mathrm{opt}} = \frac{y_{i,\mathrm{opt}} + y_{i-1,\mathrm{opt}}}{2} \qquad (i = 2,3\cdots,N) \tag{3-134}$$

$$y_{i,\mathrm{opt}} = \frac{\int_{x_{i,\mathrm{opt}}}^{x_{i+1,\mathrm{opt}}} xp(x)\mathrm{d}x}{\int_{x_{i,\mathrm{opt}}}^{x_{i+1,\mathrm{opt}}} p(x)\mathrm{d}x} \qquad (i = 1,2\cdots,N) \tag{3-135}$$

(3-134)式说明，最小均方误差量化器的判决电平应取在两相邻输出电平的中间，而(3-135)式说明，该量化器的输出电平应选择在两相邻判决电平间隔所对应的概率密度函数的重心。如果概率密度函数不是均匀分布的，那么所得到的最佳量化器是一个非均匀量化器。图 3-40(a)给出了一个非均匀量化特性的例子。



(a) 典型特性　　(b) 非线性的度量

图 3-40　非均匀量化器

量化器的非线性程度可以通过曲线 $C(x)$ 来表示[见图 3-40(b)]。图中给出了动态范围为 $[-x_{\max}, x_{\max}]$、级数为 N 的均匀量化器和非均匀量化器特性之间的对比。当 N 足够大时，$C(x)$ 的斜率近似为两个量化器的量化台阶之比，即

$$\frac{\mathrm{d}C(x)}{\mathrm{d}x} \approx \frac{2x_{\max}}{N\Delta_i} \tag{3-136}$$

3.9.3 最小熵量化器

级数为 N 的量化器的输出通常以二进制方式编码。每个样值所对应的码字的长度 L 用下式表示：

$$L \geqslant \log_2 N \quad (\text{比特 / 样值}) \tag{3-137}$$

当 N 为 2 的整数幂时，上式取等于号。

假设将量化器输出视为一个具有 N 个符号的离散无记忆信源（例如，在具有最佳预测器的 DPCM 系统中，可以足够精确地假设量化器输出是统计独立的），则该信源的熵 $H(Q)$ 为

$$H(Q) = -\sum_{i=1}^{N} P_i \log_2 P_i \tag{3-138}$$

式中，P_i 为量化器输出电平 y_i 出现的概率。P_i 由下式给出：

$$P_i = P\{Y = y_i\} = \int_{x \in A_i} p(x)\mathrm{d}x \tag{3-139}$$

$H(Q)$ 代表了信源的平均信息量。我们将会从 3.10 节中了解，利用熵编码（通常是变长的）可以将输出码字的平均码长降至接近于或等于信源的熵。这就是说，当熵编码是理想的时候，它可以将平均码长压缩到比采用二进制编码时短，二者的差值记为 ΔL，则

$$\Delta L = \log_2 N - H(Q) \tag{3-140}$$

为了使 ΔL 尽可能地大，我们将量化器的设计优化问题重新定义为：在给定 σ_d^2 的条件下，求判决电平和输出电平，使得输出熵为最小值。也就是说，我们把量化器和熵编码结合在一起来考虑它们的联合优化问题。与 3.9.2 节中最小均方误差准则下的最佳量化器相对应，这样设计的量化器称为**最小熵量化器**。

当量化级数 N 足够大，即量化台阶足够小时，我们可以假设在 x_i 和 x_{i+1} 之间 $p(x)$ 是一个常数，且 y_i 取 x_i 和 x_{i+1} 的平均值，则

$$\int_{x_i}^{x_{i+1}} (x - y_i)^2 p(x)\mathrm{d}x \approx \frac{1}{12} p(x_i)\Delta_i^3 \tag{3-141}$$

其中 $\Delta_i = x_{i+1} - x_i$ 为第 i 个量化台阶。将(3-141)式和(3-136)代入(3-132)式，得到

$$\sigma_d^2 \approx \frac{1}{12}\sum_{i=1}^{N} p(x_i)\Delta_i^3 \approx \frac{x_{\max}^2}{3N^2}\int_{-x_{\max}}^{x_{\max}} p(x)\left[\frac{\mathrm{d}C(x)}{\mathrm{d}x}\right]^{-2}\mathrm{d}x \tag{3-142}$$

同样由于 N 足够大，(3-138)式可以写成

$$H(Q) = -\sum_{i=1}^{N} p(x_i)\Delta_i \log_2 p(x_i) - \sum_{i=1}^{N} p(x_i)\Delta_i \log_2 \Delta_i \tag{3-143}$$

其中第一项近似为量化器输入信号 x 的熵 $H(X)$。将(3-136)式代入(3-143)式，得到

$$H(Q) = H(X) + \log_2 \frac{N}{2x_{\max}} + \int_{-x_{\max}}^{x_{\max}} p(x)\log_2\left[\frac{\mathrm{d}C(x)}{\mathrm{d}x}\right]\mathrm{d}x \tag{3-144}$$

根据拉格朗日乘数法，在 σ_d^2 为常数的条件下求 $H(Q)$ 的极小值问题，可以变为求下式的极值问题：

$$p(x)\log_2\left[\frac{\mathrm{d}C(x)}{\mathrm{d}x}\right] + \lambda p(x)\left[\frac{\mathrm{d}C(x)}{\mathrm{d}x}\right]^{-2} \tag{3-145}$$

其中 λ 为任意常数。令(3-145)式对 $\mathrm{d}C(x)/\mathrm{d}x$ 的导数为零，得到

$$\frac{\mathrm{d}C_{\mathrm{opt}}(x)}{\mathrm{d}x} = \left(\frac{-\lambda}{3}\right)^{\frac{1}{2}} = \text{常数} \tag{3-146}$$

在图 3-40(b)所示的 $C(x_{\max}) = x_{\max}$ 和 $C(0) = 0$ 的边界条件下，则有

$$C_{\mathrm{opt}}(x) = x \tag{3-147}$$

上述结果说明，当 N 足够大时，对于给定的量化失真，最小熵量化器是一个均匀量化器。

关于量化器和熵编码联合优化的问题，也可以从另一个角度来进行，即在给定熵 $H(Q)$ 的条件下，求判决电平和输出电平，使得量化误差 σ_d^2 为最小值。此时获得的量化器称为**熵约束最佳量化器**。从这个角度进行的优化结果表明[5]，当 N 足够大时，均匀量化器也接近于最佳。如果我们在量化器和熵编码的联合优化中，将熵 $H(Q)$ 看成“速率”（熵编码输出的每符号比特数），那么无论是在失真约束下最小化速率，还是在速率约束下最小化失真，均匀量化器都是或者接近于最佳的。因此，在图像和视频压缩编码的国际标准中，都采用了均匀量化和熵编码结合的方法。

3.9.4 自适应量化

在压缩编码中还常常使用自适应量化器。**自适应量化器**有一组预先设定好的、具有不同量化台阶的均匀量化器，在工作时根据信号特性的动态变化自动切换到相应的量化器（即切换到相应的量化步长）。例如在图像数据压缩中，根据视觉的空间域掩蔽效应，在图像细节丰富的部位使用量化步长大的量化器，在细节少的部位使用量化步长小的量化器，以使图像各部位由量化而产生的噪声的能见度大致相同。

度量图像细节的丰富程度可以有多种准则，例如，在空间域中，以相邻像素间的灰度之差的绝对值为依据、以已接收到的前几个像素预测误差值的加权和为依据，或直接以预测误差信号本身为依据等。一种频域度量方法则是，定义图像的**活动性（Activity）函数** A_F，

$$A_F = 1 + q\sum_{i=1}^{N-1} |\ y_i\ |^2 \tag{3-148}$$

式中，y_i 为第 i 次系数；q 为归一化常数；N 为变换系数的个数。由于上式的求和符号中没有包含 $i=0$ 的系数，因此 A_F 代表了图像 AC 分量的能量（即空间域亮度变化的快慢）。根据 A_F 值的大小将图像块分为几种类型，每一种类型使用不同的量化器。值得指出，多数判断准则都以解码端也同时具备的信息为依据，这样不必将量化台阶变动的情况传送给解码端，解码端就能够根据同一准则正确地解量化。

3.9.5 预测误差和 DCT 系数的量化

预测误差通常近似地遵从拉普拉斯分布。已知概率分布，根据前面介绍的方法，即可设计量化器。

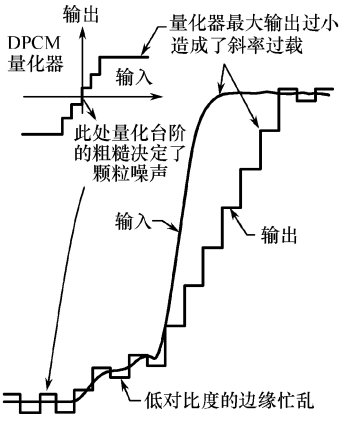


图 3-41 DPCM 的量化误差

用量化后的预测误差重建原来的信号将引入失真，典型的失真情况如图 3-41 所示。在信号较为平坦处（预测误差小），信号微小的变化使量化后的预测误差在量化器两个最小电平之间波动，造成重建信号的颗粒噪声（Granular Noise）。当信号忽高忽低时，预测误差的变动较大，如果量化粗糙的话，重建信号将产生边缘忙乱（Edge Business）。在信号的上升沿较陡（预测误差幅度很大）时，如果量化器的动态范围（最高量化电平）过小，重建信号将“赶不上”原信号的变化，这称为斜率过载（Slope Overload）。

预测误差的量化误差只引起周围邻近区域内（局部）图像的失真，而一个 DCT 系数的量化误差却对产生此系数的整块（全局）图像的质量产生影响。考虑到图像特性的空间不均匀性，通常将一幅图像划分成 8×8 或 16×16 的块，再对块图像进行变换。在图像分块做 DCT 时，直流（及低频）分量的量化误差将扩大块与块之间平均灰度的差异，从而在重建图像上显现出明显的分块结构，称为**块效应**。DCT 的高频系数在其幅度相当小时，将被“截断”（即量化到零）。高频系数的量化误差（包括因截断而产生的误差），将降

低图像的分辨率，使细节模糊；在亮度突变的附近由于吉布斯(Gibbs)效应而产生亮暗相间的条纹(Ringing)，同理在彩色突变处产生颜色的泄漏(Color Bleeding)等。在活动图像序列中，这些失真还可能造成原来被运动物体遮挡而新暴露出来的背景处出现“脏痕”。

与预测误差的量化相同，变换系数的量化器也要根据其概率分布来设计。图像信号 DC 分量的概率分布一般近似为均匀分布。至于 AC 分量的分布，虽然以前很长一段时间曾沿用高斯分布，但实际上用均值为零的拉普拉斯分布来近似更为适合[6]。

3.10 熵编码

3.10.1 熵编码的基本概念

在前面几节中我们分别讨论了预测编码和各种变换编码，这些编码都可以看成是通过解除空间或时间上的相关性，将原始信号转换成另一种形式(预测误差或变换系数)来表达。在这种新形式下，信源可以近似认为是无记忆的，即各样值之间已没有相关性，再经过量化后信源只产生有限个数的符号，因此可近似看成是一个离散无记忆信源(见图 3-42)。根据 3.2.2 节的论述可知，对于离散无记忆信源只要各事件出现的概率不相等，该信源就仍然有冗余存在，还有进一步进行数据压缩的可能性，这就是在熵编码中所考虑的问题。

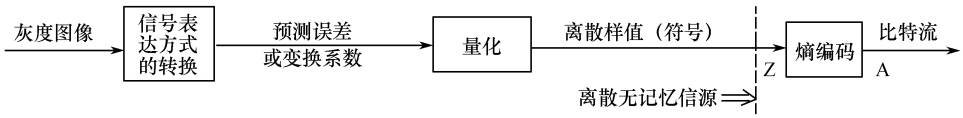


图 3-42 熵编码所处的位置

如前所述，在预测编码、变换编码、子带编码或小波变换之后，将预测误差、变换系数或子带信号量化，将造成少量信息的丢失。存在信息丢失的编码方法称为**有损编码**。本节介绍的熵编码不引起信息损失。无信息损失的编码方法称为**无损编码**。

我们以下式所表示的一个具有 4 个电平(即 4 个符号)的离散信源为例来进行讨论。

$$X=\begin{Bmatrix} s_1 & s_2 & s_3 & s_4 \\ 1/2 & 1/4 & 1/8 & 1/8 \end{Bmatrix} \tag{3-149}$$

该信源中 4 个符号出现的概率是不同的。此信源的熵为 $H(X)=1.75$ 比特/符号。如果用表 3-1 第三行所示的 2 位的二进制数码来代表这 4 种电平，例如用 10 来表示电平 s_3，这一过程叫编码，10 称为**码字**，如表 3-1 所示的符号与码字一一对应关系的集合称为**码表**或**码本**。

表 3-1 一个 4 符号信源的码表

符　　号	s_1	s_2	s_3	s_4
出现的概率	1/2	1/4	1/8	1/8
编码方式Ⅰ	00	01	10	11
编码方式Ⅱ	0	10	110	111

在第一种编码方式中(表中第三行)，每个符号所给予的码字长度都相同，平均码长 $\overline{N}=2$ 比特/符号，$\overline{N}$ 大于熵(1.75 比特/符号)。也可以改换成另一种编码方式，如表 3-1 中第四行所示。在这种编码方式中各符号所对应的码有不同的长度，称为**变长编码** VLC(Variable Length Coding)。设符号 s_i 所对应的码的字长为 $n(s_i)$ 比特，则平均码长为

$$\overline{N}=\sum_i p(s_i)n(s_i) \quad (\text{比特}/\text{符号}) \tag{3-150}$$

将第二种编码方式中各个符号的码字长及其对应的概率代入(3-150)式,对于本例而言,$\overline{N}=1.75$ 比特/符号,正好等于信源的熵。显然第二种编码方式比第一种具有更低的数据率,并且已达到了基本极限。这意味着,按照香农信息论,不可能再找到任何其他的无失真编码方式,其平均字长比这种编码方式更短。

当用二进符号表示输出码字时,相当于从一个新符号集 $A=\{0,1\}$ 中,选取符号 $a_i(i=1,2)$ 来代表原信源所输出的序列。为了达到数据压缩的目的,应使新符号集 A 的概率模型尽可能接近于等概率分布。也就是说,使信源 A 的熵最大,每个符号 a_i 平均携带的信息量最多,从而表达原信源输出所需要的符号 a_i 数目最少。这从下面的例子中可以得到证实。当采用表 3-1中的第 2 种编码方式时,原输出序列与编码后的序列有如下关系:

	原信源输出序列	s_1	s_2	s_1	s_3	s_2	s_1	s_1	$s_4\cdots$
方式 Ⅱ	编码后的序列	0	10	0	110	10	0	0	111…
方式 Ⅰ	编码后的序列	00	01	00	10	01	00	00	11…

(3-151)

可以看出,用方式Ⅱ编码后的序列中 0 和 1 出现的重复频率相同,因此这种编码方式能够给出较低的数据率。在(3-151)式中,编码方式Ⅱ只用 14 个符号,而编码方式Ⅰ需用 16 个符号表示同一个原始信源序列。

如果编码符号集 A 中符号的个数为 M,经编码以后可能达到的最大熵值则为 $\log_2 M$。若原信源的熵为 $H(X)$,经编码后的平均码长为 $\overline{N}$,编码符号集中一个符号所携带的平均信息量则为 $H(A)=H(X)/\overline{N}$。显然

$$\frac{H(X)}{\overline{N}}\leqslant\log_2 M \tag{3-152}$$

我们将**编码效率** η 定义为

$$\eta=\frac{H(X)}{\overline{N}\log_2 M} \tag{3-153}$$

当 $\eta=1$ 时,有

$$\overline{N}_{\min}=\frac{H(X)}{\log_2 M} \tag{3-154}$$

这就是在编码中可能达到的最小平均码长的极限。当 M 取 2 时,例如 $A=\{0,1\}$,最小平均码长则等于原信源的熵。熵编码的宗旨在于找到信源符号集内的符号与码字的一一对应关系,以使平均码长达到上述极限。不难看出,这种编码方式是无失真的。

实际应用中的熵编码方式主要有霍夫曼编码和算术编码两种,下面分别进行介绍。

3.10.2 霍夫曼编码

霍夫曼(Huffman)编码的基本思想是,对出现概率较大的符号 s_i 取较短的码长,而对概率较小的符号则取较长的码长,因此它是一种变长码。霍夫曼码通常被称为最优码。最优的含义是,对于给定的符号集和概率模型,找不到任何其他整数码比霍夫曼码有更短的平均字长。所谓整数码是指每个符号所对应的码字的位数都是整数。

现在通过 $M=2$ 的例子来介绍霍夫曼码的编码过程。设原信源为

$$X=\begin{Bmatrix}x_1 & x_2 & \cdots & x_m\\ p_1\geqslant & p_2\geqslant & \cdots & \geqslant p_m\end{Bmatrix} \tag{3-155}$$

其编码的具体步骤如下。

(1) **合并**:将信源中最小概率的两个事件合并成一个事件。由于这些事件是统计独立的,合并后的事件出现的概率等于原来两个事件的概率和。

(2) **置换**:将合并后的信源的事件重新按概率大小排列(概率相等的事件可按任意顺序排列)。

(3) 若合并后的信源中事件的个数大于 2,重复执行步骤(1)和(2);若合并信源中事件个数等

于 2，则进行步骤(4)。

(4) **赋值**：在最后得到的信源

$$X^{m-2}=\begin{Bmatrix} x_1^{(m-2)} & x_2^{(m-2)} \\ p_1^{(m-2)} \geqslant & p_2^{(m-2)} \end{Bmatrix} \tag{3-156}$$

中，给 $x_1^{(m-2)}$ 和 $x_2^{(m-2)}$ 分别赋值 0(或 1)和 1(或 0)。

(5) **反推**：由于 $x_2^{(m-2)}$(或 $x_1^{(m-2)}$)一定是由两个事件构成的，因此，对应于这两个事件的码字由 $x_2^{(m-2)}$ 的码字分别加上 0 和 1 构成。以此类推，一直到返回原信源符号为止。

现在利用图 3-43 来说明上述编码过程。假设构成信源的符号以及它们各自出现的概率为

$$X=\begin{Bmatrix} s_1 & s_2 & s_3 & s_4 & s_5 \\ 1/3 & 1/4 & 1/5 & 1/6 & 1/20 \end{Bmatrix} \tag{3-157}$$

图 3-43 中从左到右用虚线标出的 4 个部分表示 4 个合并与置换的步骤。在第 1 步中，概率最小的 s_4 和 s_5 合并，其合成事件的概率为 13/60。由于 1/5<13/60<1/4，因此合成事件在第 2 步中排在 s_2 和 s_3 之间。以此类推，在第 4 步中，只剩下了两个事件，合并过程结束。然后给每一个节点(图中○点处)的两个分支分别赋以 0 和 1。如图所示的树状结构称为**码树**。从树根(图中用箭头指出的节点)到左端树梢的各个分支所经过的码串接起来就构成该树梢上符号所对应的码字。例如，从树根经过两个节点到 s_3 所构成的码为 11。

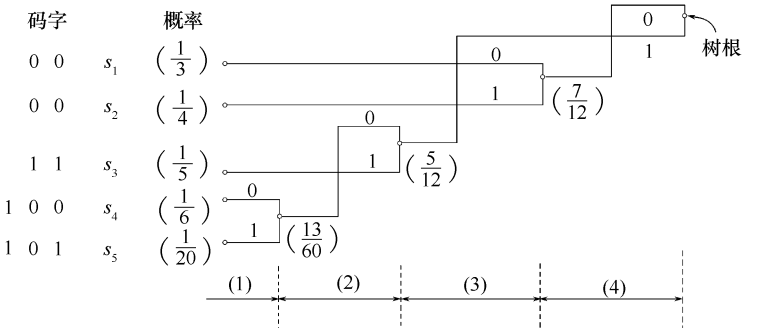


图 3-43　霍夫曼编码过程示意图

由于一个节点的上下两个分支既可赋值 0(或 1)，也可赋值 1(或 0 值)，而且对概率相同的两个事件的排列顺序也是任意的，因此对同一信源所构成的霍夫曼码并不是唯一的，但是它们的平均字长却是一样的。

这样构造的霍夫曼码是无歧义的。所谓**无歧义**是指在译码过程中，对某个码字的解释是唯一的。现在举一个例子来说明。对于表 3-1 所给出的信源概率模型，读者根据霍夫曼码的构码过程，可以证明表中给出的第二种码即为霍夫曼码。假设该信源输出序列和编码后的序列如(3-151)式所示，当接收端收到码流 01001101000111…时，按照码表，以 0 开头的码字只有 0，因此解出第一个符号为 s_1；除去已解的码 0 以后，码流则剩下 1001101000111，码表中以一个 1 开头的码只有 10，因此得到第二个符号 s_2；除掉 10 之后，码流剩下 01101000111，第一个码 0 决定了译码结果只能是 s_1；在剩余码流 1101000111 中，开头的两个码为 11，而码表中 2 比特的码字只有 10，因此可以断定应该按 3 位来译码，得到 s_3；以此类推。

由以上译码过程可以看出，虽然霍夫曼码是变长的，码流中又没有分隔码字的标识符，但由于它独特的**前缀(Prefix)特性**，即短码字永远不会成为长码字的开头，因此，完全能够从码流直接恢复原信源所输出的符号序列来。

值得注意的是：①由于霍夫曼构码过程的最基本依据是信源的概率分布，如果信源的实际概率模型与构码时所假设的概率模型有差异，实际的平均码长将大于预期值，编码效率下降。如果差异

很大，则有可能比使用定长码（如表 3-1 中的第一种编码方式）的平均字长还要长。对于这种情况，解决的办法是更换码表，使之与实际概率模型相匹配；或者直接使用定长码。②霍夫曼码对传输错误十分敏感，如果码流中出现 1 比特错误，不仅对应的码字会译错，而且由于前缀特性被破坏，可能使后续的译码在错误的码字边界上进行，从而使错误传播下去。

3.10.3 算术编码

算术编码是另一种能够趋近于熵极限的最佳编码方式，它与霍夫曼编码一样，也是对出现概率较大的符号采用短码，对概率较小的符号采用长码。但是它的编码原理却与霍夫曼编码很不相同，也不局限于非使用整数码不可，这使得它比霍夫曼码有更高的效率。

1. 算术编码的基本原理

假设一个具有 4 个电平的信源，其概率模型如表 3-2 所示。把各符号出现的概率表示在如图 3-44(a)所示的单位概率区间之中，区间的宽度代表概率值的大小。由图可以看出，各符号所对应的子区间的边界值，实际上是从左到右各符号的累积概率。在算术编码中通常采用二进制的分数来表示概率，表 3-2 和图 3-44(a)中都标出了相应的二进概率数值。每个符号所对应的概率区间都是半开区间，即该区间包括左端点，而不包括右端点，如 s_1 对应[0,0.001)，s_2 对应[0.001,0.011)，等等。

表 3-2　一个 4 符号信源的概率模型

符　　号	s_1	s_2	s_3	s_4
概率(十进制)	1/8	1/4	1/2	1/8
概率(二进制)	.001	.01	.1	.001
累积概率	0	.001	.011	.111

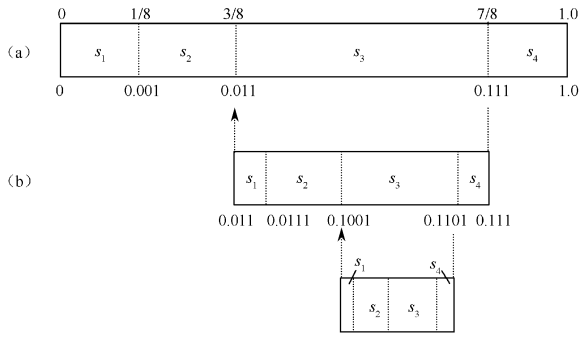


图 3-44　算术编码过程

现在以符号序列 $s_3s_3s_2s_4\cdots$ 为例来解释编码过程。算术编码所产生的码字实际上是一个二进制分数值的指针，该指针指向所编的符号对应的概率区间。按照这一法则，序列的第一个符号为 s_3，我们用指向图 3-44(a)中第三个子区间的指针来代表这个符号。从原理上来讲，指针指向[0.011,0.111)区间内的任何部位都可代表 s_3，但通常约定为指向它的左端点，由此得到码字 0.011。后续的编码将在前面编码指向的子区间内进行。将[0.011,0.111)区间再按符号的概率值划分成 4 份，如图 3-44(b)所示。对第二个符号 s_3，指针指向 0.1001(s_3 区间的左端)，也就是码字串变为 0.1001。然后如图(b)所示，s_3 所对应的子区间又被划分为 4 份，开始对第三个符号进行编码。余下的步骤以此类推。

总结上述划分子区间的递归过程，可将算术编码的基本法则归纳如下：

(1)初始状态

编码点(指针所指处)$C=0$

区间宽度 $A=1.0$

(2)新编码点 $C=$原编码点 $C+$原区间 $A\times P_i$　　(3-158)

新区间 $A=$原区间 $A\times p_i$　　(3-159)

其中 p_i 和 P_i 分别为所编符号 s_i 对应的概率和累积概率。

根据上述法则，对序列 $s_3 s_3 s_2 s_4 \cdots$ 进行编码的过程如下。

第一个符号(s_3)：

$$C = 0 + 1 \times 0.011 = 0.011$$
$$A = 1 \times 0.1 = 0.1$$

第二个符号(s_3)：

$$C = 0.011 + 0.1 \times 0.011 = 0.1001$$
$$A = 0.1 \times 0.1 = 0.01$$

第三个符号(s_2)：

$$C = 0.1001 + 0.01 \times 0.001 = 0.10011$$
$$A = 0.01 \times 0.01 = 0.0001$$

第四个符号(s_4)：

$$C = 0.10011 + 0.0001 \times 0.111 = 0.1010011$$
$$A = 0.0001 \times 0.001 = 0.0000001$$

该码字串 0.1010011 用 $7b$ 表示了 4 个符号，平均码长为 $7/4=1.75b$/符号。

在解码器中，当收到码字串 0.1010011 时，由于这个码字串指向子区间[0.011，0.111)[见图 3-44(a)]，因此，解出的第一个符号应为 s_3。然后，采取与编码过程相反的步骤，即从码字串中减去已解符号子区间的左端点的数值(累积概率)，并将差值除以该子区间的宽度(概率值)，则得到新码字串(0.1010011－0.011)÷(0.1)＝0.100011。由图 3-44(a)中可以看出，新码字串仍落在[0.011，0.111)区间之内，因此，解出的第二个符号仍为 s_3。后面的过程可以此类推。

在算术编码中，一个值得注意的问题是**进位**。在霍夫曼编码中，如(3-151)式所示，后续符号的码字只是简单地附加到已编好的码字串之尾，并不改变已有的码字串。而在算术编码中则不同，如在上面的例子中，编完第三个符号之后得到的码字串为 0.10011，在对第四个符号编码时，将码串的前 3 位由 0.100 变成 0.101，这便是由于相加运算中的进位引起的。

2. 二进制算术编码

二进制算术编码是一种常用的算术编码方法，所谓二进制是指输入的字符只有两种。如果信源字符集内包含有多个字符，则先将这些字符经过一系列的二进判决，变成二进制字符串。图 3-43 所示的霍夫曼树就是这样的判决方法之一(不遵从霍夫曼概率置换原则的二进树也可以)。在那里，字符 s_4 的转换过程可以分解成 3 次二进判决(100)。

在二进制算术编码器的两个输入字符中，出现概率较大的一个通常称为 **MPS**(More Probable Symbol)，另一个称为 **LPS**(Less Probable Symbol)。假设 LPS 出现的概率为 Q_e，MPS 出现的概率则为$(1-Q_e)$，图 3-45 表示出这两个符号所对应的概率区间。

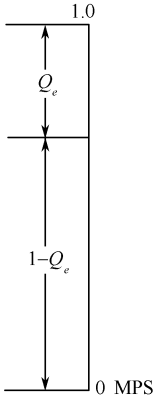


图 3-45　二进概率区间

对这两个符号构成的序列的编码与前面介绍的算术编码的基本原理是相同的，仍然是不断划分概率子区间的递归过程。如果我们规定编码指针指向子区间的底部(见图 3-45)，那么编码规则为：

对于 MPS

$$C = C \tag{3-160}$$
$$A = A(1-Q_e) = A - AQ_e \tag{3-161}$$

对于 LPS

$$C = C + A(1-Q_e) = C + A - AQ_e \tag{3-162}$$
$$A = AQ_e \tag{3-163}$$

在霍夫曼编码中，最短的码字为 1 比特，所以即使在对最常出现的符号进行编码时，也需要在前面已编好的码字串上再增加 1 比特。而在算

术编码中，由(3-160)式可以看出，对 MPS 编码不增加已编好的码字串的长度。这是算术编码比霍夫曼编码优越的地方。

在具体实现上述算法时，如下几个问题值得注意：

① 在概率子区间不断划分的过程中，区间宽度 A 越来越小，因此，用来表示 A 的数字的位数则需要越来越多；

② 完成(3-161)式至(3-163)式的运算都要用到成本较高(无论是用硬件实现还是用软件实现)的乘法运算；

③ 当已编好的码字串中连续出现多个 1 时，若后续编码过程中在最后一位上加 1，将连续改变前面已编好的码字，产生连续多个 0，直到出现 1 为止，这就是前面提到过的进位问题。

为了有效地实现编码算法，人们提出了若干办法来解决上述问题，从而构成了不同类型的二进制算术编码器，如 Q 编码器[20]、QM 编码器[21]和 MQ 编码器[19]等。下面我们仅简单介绍其中的 **QM 编码器**。

用有限精度的算术运算来计算概率区间 A，可以保证 A 的有效数字的位数不至于随码字串的增加而增加。例如 $A=0.001$ 可以用 1.0×2^{-3} 来表示，当 A 被分割到更小值时，则增加指数值，而有效数字的位数仍保持在规定的范围之内。假设这个范围选择在 0.75 到 1.5(十进制数字)之间，当 $A<0.75$ 时，将 A 乘以 2(相当于 2 的负指数增加 1)，使有效数字保持在 0.75～1.5 之间，这样，仍然可以用原来的位数表示，这无论对用硬件还是用软件来实现编码都是有利的。上述过程称之为**重新归一化**。乘 2 运算可以由将数据在寄存器中左移一位的操作来完成。显然，当区间 A 重新归一后，码字串 C 也应当随之归一，否则 C 将不能指向正确的区间。

当 $1.5>A\geqslant0.75$ 时，$AQ_e\approx Q_e$。利用这个近似式，可将(3-161)式至(3-163)式简化而无需做乘法。此时二进概率区间变成如图 3-46 所示。Q_e 一般比较小，只有当 Q_e 接近于 0.5 的情况下，上述近似式引入的误差才比较明显。

当 Q_e 接近 0.5 时，由此近似产生的误差有可能发生 LPS 区间大于 MPS 区间的情况，图 3-47 给出了一个例子。图(a)中 $Q_e=0.5$，初始的 $A=1.35$，MPS 和 LPS 的区间分别为[0.0，0.85)和[0.85，1.35)。在下一个编码周期中(图(b))，区间 A 更新为 $A=A-Q_e=0.85$，LPS 区间 $Q_e=0.5$，MPS 区间变成 $A-Q_e=0.35$。在这种情况下，即 $Q_e>A-Q_e$ 时，需要将 MPS 和 LPS 符号及区间互换，这称为**条件互换**(Conditional exchange)。图(b)表示出了条件互换后的结果。在需要进行条件互换时，由于 $Q_e\leqslant0.5$，我们有 $0.5\geqslant Q_e>A-Q_e$，此时，两个子区间(Q_e 和 $A-Q_e$)都小于 0.75，肯定需要重新归一化 A 和 C。因此，条件互换总是放在编码器检测到需要重新归一化之后再进行。

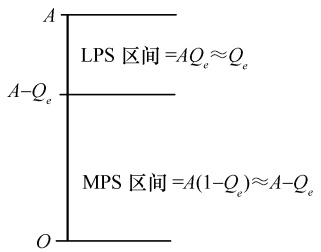


图 3-46　QM 编码器的二进区间

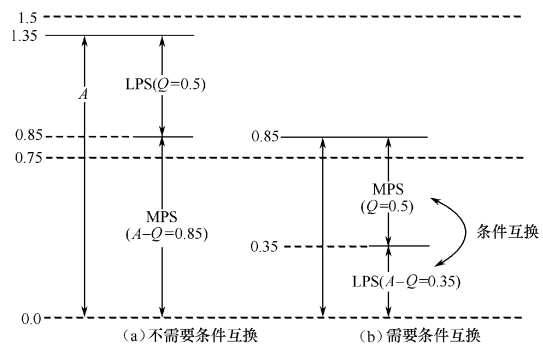


图 3-47　条件互换

将重新归一化和条件互换结合进(3-160)式至(3-163)式所表示的编码过程，得到 QM 编码器的编码算法为：

对于 MPS：

```
Begin
  C=C;
  A=A-Qe;
  If(A<0.75)then
    If(A<Qe)then
      C=C+A;
      A=Qe;
    Endif
    重新归一化 A 和 C;
  Endif
End
```

对于 LPS：

```
Begin
  A=A-Qe;
  If(A≥Qe)then
    C=C+A;
    A=Qe;
  Endif
  重新归一化 A 和 C;
End
```

相应的解码算法为：

```
Begin
  If(C>(A-Qe))then
    解码 LPS;
    C=C-(A-Qe);
    A=Qe;
    Else
      解码 MPS;
      A=A-Qe;
  Endif
  If(A<0.75)then
    重新归一化 A 和 C;
  Endif
End
```

为了简单起见，这里没有考虑条件交换。

关于进位的问题，可以用在编码器输出之前设置一个缓存器的办法来解决。连续出现 1 的码字串（如连续的 8 个 1）先不输出，而将它送入一个堆栈暂存，若后续编码引起进位，则改变堆栈中的数据后再输出。若不引起进位，则直接输出堆栈中的码字。

与霍夫曼编码一样，如果输入信号的实际概率模型（二进编码中的 Q_e 值）与假设值不一样，算术编码的效率也要下降。为了提高编码效率，可以通过一定方式实时地对实际输入信号的 Q_e 值进行估计，编码器则根据所估值动态地更新 Q_e，这就构成了所谓的**自适应二进制算术编码**[19,21]。在 QM 编码中，对概率的估值是通过一个预先设定的 Q_e 表来实现的。该表包括一组按大小排序的 Q_e 值，当编码过程进行重新归一化时，若因编码 LPS 而导致归一化，Q_e 更新为表中下一个较低的值；若因编码 MPS 而导致归一化，Q_e 则更新为相邻的较高的值。

习　题　三

1. 设离散信源输出两个符号的序列，这两个符号从符号集 $A=\{0,1\}$ 中随机地选取，并且 $P(1)=0.8$，$P(0)=0.2$，
 (1) 若这两个符号的条件概率为 $P(0/1)=0.1$ 和 $P(1/0)=0.4$，求该信源的序列熵；
 (2) 若该信源是无记忆的，求该信源的序列熵，并与(1)的结果进行比较。
2. 设具有 4 个等级的量化器的输出序列为{1,1,3,4,2,1,2,1,3,1,1,2,4,2,1,1}，
 (1) 按无记忆信源计算其熵；

(2) 按一阶马尔柯夫信源计算其熵，并与(1)的结果进行比较。

(提示：马尔柯夫信源熵等于条件熵的加权平均，即熵以信源所处的某个状态为条件，而加权值等于此状态的概率。)

3. 设 $x(n)$ 的傅氏变换 DFT 为 $X(e^{j\omega})$，

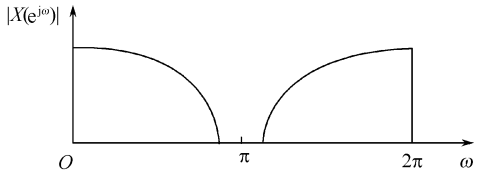
(1) 证明 $x(n)$ 经 2∶1 抽取后的频谱为 $\frac{1}{2}[X(e^{j\omega/2})+X(-e^{j\omega/2})]$；

(2) 证明 $x(n)$ 经 1∶2 内插后的频谱为 $X(e^{j2\omega})$；

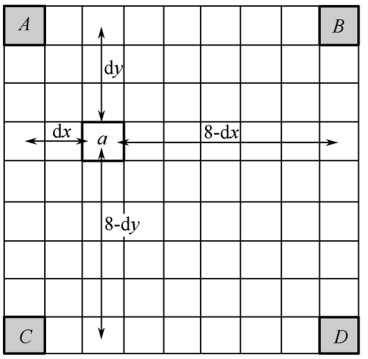
(3) 若 $|X(e^{j\omega})|$ 如图所示，画出经抽取和内插后的幅度谱示意图。

4. 求(3-32)式和(3-33)式所示的内插滤波器的传输函数 $H(e^{j\omega})$，并用 Matlab 画图。根据图比较这两种方法的优劣。

5. 将线性内插方法扩展到二维信号，就得到**双线性内插方法**。它利用相邻 4 个像素(如下图中深色方块 A，B，C，D)内插出 4 者之间区域内任意点(如下图中 a)上的数值，具体做法是：在一个方向(如行方向)上利用线性内插分别求出与 a 同列、与 A、B 同行和与 a 同列、与 C、D 同行的两个像素值；再将这两个像素值在另一个方向(列方向)上通过线性内插求出 a 的数值。根据上述思想，写出用 A、B、C、D 表示的 a 的内插值。



习题 3 图



习题 5 图

6. 为什么内插滤波器和抗混叠滤波器通常都使用 FIR 滤波器，而不是 IIR 滤波器？

7. 用公式说明为什么图 3-15 与图 3-13 相比，解码器输出更接近于编码器输入(残存量化误差小)？

8. 在具有运动补偿的帧间预测编码中，如果运动矢量估计得不准确产生的后果是什么，会不会引起解码后的图像失真，为什么？

9. 根据图 3-21 所示，请说明

(1) 初始(第一步)的搜索点是如何确定的？

(2) 第二步的搜索点是如何确定的？

(3) 为什么选择左上方点进行第 3 步搜索？

(4) 搜索是如何中止的？是否使用了早期中止策略？

10. (1) 若 $\boldsymbol{\Sigma}$ 是对称实数矩阵，且 $\boldsymbol{\phi}$ 是 N 维矢量，证明 $\nabla_{\boldsymbol{\phi}}[\boldsymbol{\phi}^{\mathrm{T}}\quad \boldsymbol{\Sigma}_X\quad \boldsymbol{\phi}]=2\boldsymbol{\Sigma}_X\boldsymbol{\phi}$ 和 $\nabla_{\boldsymbol{\phi}}[\boldsymbol{\phi}^{\mathrm{T}}\boldsymbol{\phi}]=2\boldsymbol{\phi}$；

(2) 利用上述结论，在 MMSE 准则下，证明(3-61)式。

11. 设有一个信源矩阵 $\boldsymbol{X}$ 如图所示。

(1) 用下述正交矩阵进行变换，称为**哈达玛变换**。求上述信源矩阵的变换系数矩阵 $\boldsymbol{Y}$；

5	11	8	10
9	8	4	12
1	10	11	4
19	6	15	7

习题 11 图

$$\boldsymbol{Y}=\frac{1}{2}\begin{bmatrix}1&1&1&1\\1&1&-1&-1\\1&-1&-1&1\\1&-1&1&-1\end{bmatrix}\cdot\boldsymbol{X}\cdot\begin{bmatrix}1&1&1&1\\1&1&-1&-1\\1&-1&-1&1\\1&-1&1&-1\end{bmatrix}$$

(2) 根据以上结果说明哈达玛变换是否具有能量集中特性，能否用于数据压缩；

(3) 写出哈达玛反变换的公式。

12. 已知信源协方差矩阵为

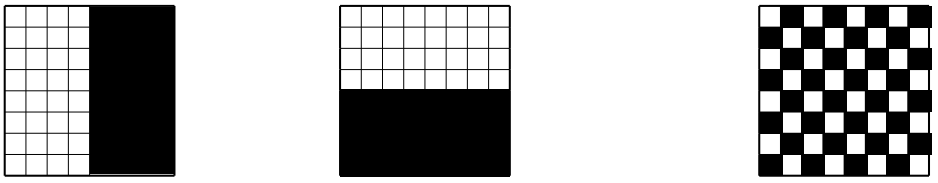
$$\boldsymbol{\Sigma}_X = \begin{bmatrix} 1 & 0 & 1 \\ 0 & 1 & 0 \\ 1 & 0 & 1 \end{bmatrix}$$

试计算最佳变换的变换矩阵 $\boldsymbol{T}$，并求信号经变换后在变换域的协方差矩阵 $\boldsymbol{\Sigma}_Y$ 。

13. 用 C 或 Matlab 编程计算下列图像块的 DFT 和 DCT。从数据压缩的角度，对这两个变换进行比较。

$$\begin{bmatrix} 16 & 11 & 10 & 16 & 24 & 40 & 51 & 61 \\ 12 & 12 & 14 & 19 & 26 & 58 & 60 & 55 \\ 14 & 13 & 16 & 24 & 40 & 57 & 69 & 56 \\ 14 & 17 & 22 & 29 & 51 & 87 & 80 & 62 \\ 18 & 22 & 37 & 56 & 68 & 109 & 103 & 77 \\ 24 & 35 & 35 & 64 & 81 & 104 & 113 & 92 \\ 49 & 64 & 78 & 87 & 103 & 121 & 120 & 101 \\ 72 & 92 & 95 & 98 & 112 & 100 & 103 & 99 \end{bmatrix}$$

14. 对图中所述三种 8×8 的亮度图形进行 DCT 变换后，估计（不必实际计算）在系数矩阵的哪些位置上，变换系数有较大数值。



(a)　　(b)　　(c)

习题 14 图

15. 说明第 2 章习题 4 图中两个正弦光栅的 DCT 频率谱是否相同？解释原因。

16. 指出图 3-34 中的每一个步骤与 3.7.3 节中的哪一个（或哪几个）公式相对应，从而说明在 TDAC 编码过程中混叠失真是如何抵消的。

17. 设一图像经图 3-37 所示的多层小波分解后，其系数如图所示。按照 EZW 算法，

(1) 确定初始门限值，并进行第 1 次主扫描，得到显著系数表；

(2) 进行第 1 次辅扫描，得到加细量化表；

(3) 缩小门限值，继续进行上述两步扫描，直至门限值为 1。写出最后得到的编码码流。

62	−34	49	10	7	13	−12	7
−31	23	14	−13	3	4	6	−1
15	14	3	−12	5	−7	3	9
−9	−7	−14	8	4	−2	3	2
−5	9	−1	47	4	6	−2	2
3	0	−3	2	3	−2	0	4
2	−3	6	−4	3	6	3	6
5	11	5	6	0	3	−4	4

习题 17 图

18. 设有一离散无记忆信源

$$X = \begin{Bmatrix} S_1 & S_2 & S_3 & S_4 & S_5 & S_6 \\ 0.3 & 0.2 & 0.2 & 0.1 & 0.1 & 0.1 \end{Bmatrix}$$

(1) 计算该信源的熵；

(2) 为该信源构造一霍夫曼码，并计算平均码长及编码效率；

(3) 为什么该码的编码效率小于 1？试总结在什么情况下，霍夫曼码的编码效率能够达到 1？（提示：符号 S_i 携带的信息量是多少？表示 S_i 的码字的码长是多少？）。

19. 根据下述码表产生的码字序列，在译码时是否是无歧义的？说明原因。

y_1	y_2	y_3	y_4	y_5	y_6	y_7	y_8
010	0001	0110	1100	00011	00110	11110	101011

20. 假设有下表所示的5种运动矢量，其出现的概率如表所示。试用一般算术编码对 MV 序列{－2，－1，0，2}进行编码，

(1) 给出编码过程和最后输出的码字(允许最后一位不能正确解码)；

(2) 输出码字需要用几比特表示？编码效率为多少？

(3) 同一序列若采用霍夫曼编码，编码效率为多少？

MV	概率(十进制)
－2	0.1
－1	0.2
0	0.4
1	0.2
2	0.1

参 考 文 献

[1] C. E. Shannon，"A mathematical theory of communication，"Bell Sys. Tech. J. ，1948，pp. 379-423，623-656

[2] H. G. Musman，et. al. ，"Advances in Picture Coding，"Proc. IEEE，Vol. 73，1985，pp. 523-548.

[3] T. Berger，"Rate-Distortion Theory"，Prentice-Hall，1971

[4] 周炯槃．信息理论基础．北京：人民邮电出版社，1983

[5] Y. Wang，et al，Video Processing and Communications，Prentice-Hall，2002

[6] A. Netravali and B. Haskell，Digital Pictures，Representation，Compression and Standards，Plenum，New York，1995

[7] P. I. Hosur and K，K. Ma，"Motion vector field adaptive fast motion estimation，"ICICS'99，pp. 7-10

[8] A. M. Tourapis，et al. ，"New results on zonal based motion estimation algorithms-advanced predictive diamond zonal search，"Proc. IEEE ISCAS'01，Vol. 5，pp. 183-186

[9] A. M. Tourapis，"Enhanced predictive zonal search for single and multiple frame motion estimation，"Visual communications and image processing 2002，Proc. SPIE，vol. 4671，PP. 1069-1079

[10] [美]N. 阿罕麦德，K. R. 罗著．数字信号处理中的正交变换．胡正名，陆传赉译．北京：人民邮电出版社，1979

[11] W. Chen，et al. "A fast computational algorithm for the discrete cosine transform，"IEEE Trans. Communications，COM-25，1977，pp. 1004

[12] C. Loeffler et al. ，"Practical fast 1-D DCT algorithm with 11 multiplications，"Proc. IEEE ICASSP'89，1989，Vol. 2，pp. 988-991

[13] B. G. Le et al. ，"FCT——A fast cosine transform，"Proc. IEEE ICASSP，1984，Vol. 2，28A. 3. pp. 1-4

[14] R. E. Crochiere，"SuB-Band Coding，"Bell Sys. Tech. J. ，Sept. 1981，pp. 1633-1654

[15] R. E. Crochiere and L. R. Rabiner、Multirate Digital Processing、Prentice Hall，1983

[16] S. Mallat，"A theory for multiresolution signal decomposition：the wavelet representation，" IEEE Trans. PAMI，Vol. 11，No. 7，1989，pp. 674-693

[17] J. M. Shapiro，"Embeding image coding using zerotrees of wavelet coefficients"IEEE Trans. SP，Vol. 41，No. 12，1993，pp. 3445-3462

[18] A. Said and W. A. Pearlman，"A new，fast and efficient image codec based on set partitioning in hierarchical trees，"IEEE Trans. CSVT，Vol. 6，No. 3，1996，pp. 243-250.

[19] [美] D. S. Taubman and W. Marcellin 著．JPEG2000 图像压缩基础、标准和实践．魏江功等译．北京：电子工业出版社，2004

[20] W. B. Pennebaker，et al. ，"An Overview of the Basic Principles of the Q-Coder Adaptive Binary Arithmetic Coder，"IBM J. Res. Develop. ，1988，Vol. 32，No. 6，pp. 717-726

[21] W. B. Pennebaker and J. L. Mitchell，JPEG：Still Image Data Compression Standard，Chapmann & Hall，1993

第 4 章　视频数据的压缩编码

4.1　基于帧的视频编码

本节介绍基于帧的视频编码和解码。这类编、解码方法以视频帧为基本编码单元，综合利用了第 3 章中介绍的第一代压缩编码技术，并为多数现行的国际标准所采用。在 4.1.1 和 4.1.2 节中我们介绍基本的编、解码方法；在 4.1.3～4.1.6 节中介绍各种提高编码性能的优化技术；而 4.1.7 节则给出典型的编码码流的层次结构。

4.1.1　典型的编码器与解码器

一个典型的视频编码器和解码器方框图如图 4-1 所示，它包括具有运动补偿的帧间预测、DCT 变换、量化、熵编码，以及与固定速率的信道相适配的速率控制等几个部分，用以在保证图像有满意的质量的前提下，最大限度地压缩码率。由于图 4-1 所示的编码器采用了多种压缩编码技术，所以常被称为**混合(Hybrid)编码器**。

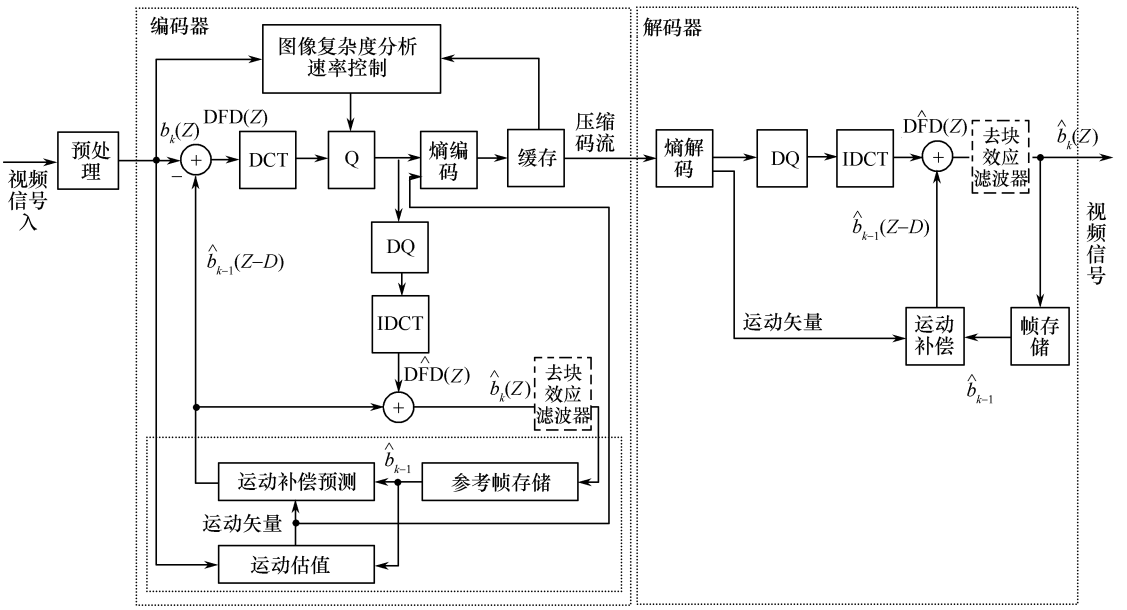


图 4-1　典型的视频编码器和解码器

对视频图像的压缩编码通常是分块进行的。如图 4-2(a)所示，将一帧图像首先分成若干**条**(Slice)，或称**片**，每条又分成许多块。对于 4∶2∶0 格式来说，亮度图像块所包含的像素数目一般为 16×16，对应的色度块像素数为 8×8。一个亮度块及其对应的两个色信号块构成一个基本处理单元，称为**宏块**(Macroblock)，标为 **MB**[见图 4-2(b)]。每一条所包含的宏块数可以是不相同的，每一帧划分的条数也视具体应用而定。

现在来讨论编码器的基本工作原理。

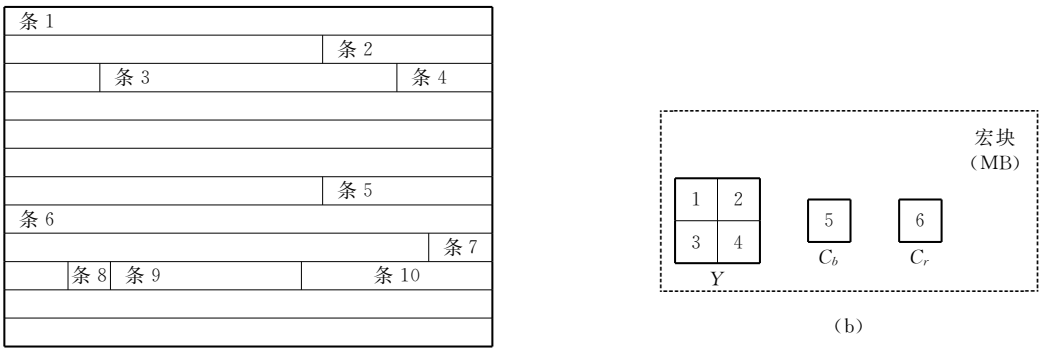


图 4-2　条与宏块

1. 图像信号的预处理

图像信号的预处理过程如图 4-3 所示。

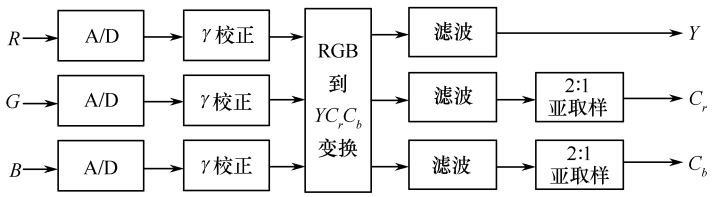


图 4-3　图像信号的预处理

模拟 R、G、B 信号首先经过 8 比特或 10 比特的 A/D 变换变成数字信号，然后分别进行 γ 校正。根据 2.2.4 节介绍的内容将 R、G、B 信号转换成亮度信号和色差信号，并按(2-24)式至(2-26)式得到 Y、C_r 和 C_b。我们在这里和以后都将 Y_d 简称为 Y。

由于人眼对色差信号高频分量的敏感程度比对亮度的低，在信号变换到 Y、C_b 和 C_r 空间之后可以用数字滤波器将 C_b、C_r 的带宽限制到 Y 信号带宽的 1/2。然后对于 4∶2∶0 的格式，在垂直和水平方向上进行 2∶1 的抽取，这样得到的色信号矩阵的样点数只是 Y 矩阵的 1/4。

2. 运动估值/补偿

为了消除帧间冗余信息，编码器采用了具有运动补偿的帧间预测(见图 4-1 下部的虚线框)。在运动估值部分，当前帧的亮度图像 b_k 被分成 16×16 的块，然后利用块匹配方法在参考帧中找到每个块 $b_k(\boldsymbol{Z})$ 的最佳匹配块，从而估计出每个块的位移矢量值。在匹配过程中，参考帧内的搜索范围根据不同的应用而定，例如可以取 $\pm 128(H)\times\pm 32(V)$，这里的 H 和 V 分别表示垂直方向和水平方向。当参考帧为前一帧 b_{k-1} 时，进行的是前向预测；参考帧为后一帧 b_{k+1} 时，则进行的是后向预测。关于何时进行前向预测、何时进行后向预测，下面将要详细讨论。

利用每个块的位移矢量 $\boldsymbol{D}$，可以从参考帧得到经过运动补偿的当前块的预测值 $\hat{b}_{k-1}(\boldsymbol{Z}-\boldsymbol{D})$。在当前块和预测块的差值图像(或称为位移后的帧间差 DFD)中，时间冗余信息已经大大减少，但还可能保留着空间冗余信息 。为了降低空间冗余度，将在后续步骤中对 DFD 进行 DCT 变换和量化。运动估值中用到的参考帧是在经过反量化和反 DCT 恢复出 $D\hat{F}D(\boldsymbol{Z})$ 之后，与 $\hat{b}_{k-1}(\boldsymbol{Z}-\boldsymbol{D})$(或 $\hat{b}_{k+1}$)相加得到的。上述带有“^”的符号表示该量引入了量化误差。同时，在 3.5.2 节中介绍块匹配方法时已经讲过，位移矢量必须伴随着 DFD 送到接收端，接收端才能从已接收帧中恢复出当前帧来。图中标出此支路直接进入熵编码模块。

按上述方法编码的块通常称为**帧间编码块**(Inter MB)。但是在有些情况下，如场景切换时，帧间预测误差很大，此时图像块可以不进行预测编码，而直接进行 DCT 变换和熵编码，这样的块称

为**帧内编码块**(Intra MB)。在采用帧间预测编码的一幅图像中,如果某一块的预测误差很大,对该块也可以采用帧内编码。对于帧内编码块,可以看成其预测值为 0。

3. DCT 变换

在 3.6 节中讲过,图像经 DCT 变换以后,DCT 系数之间的相关性已经很小,而且大部分能量集中在少数的系数上。只传送这部分系数给收端,既可以降低数据率,而又不至于使图像有明显的损伤。从原理上讲我们可以对整幅图像进行 DCT 变换,但由于图像各空间位置上细节的丰富程度不同,这种整体处理的方式效果不好。为此,采用了将宏块再分成 8×8 的子块、对每个子块进行 DCT 的办法。这种处理方法称为**块 DCT**。位于细节丰富区的子块,可以保留较多的 DCT 系数;而细节不丰富的子块,则保留较少的系数。采用块 DCT 的办法不仅便于每个子块根据各自的图像特点进行独立处理,而且降低了运算量和对存储空间的要求。

对色信号的处理与上面讨论的对亮度信号的处理方法类似,只是对色信号 C_b、C_r 不做运动估值,而是直接使用同一宏块中的亮度块的位移矢量作为色信号子块的位移矢量。根据这些位移矢量对 C_r 和 C_b 进行具有运动补偿的帧间预测,然后对预测差值图像进行块 DCT 变换。

4. 量化

在量化之前,所有的处理都没有涉及到信息的损失(除了在 DCT 变换中,系数值受有限字长影响而引入的舍入误差外),只是改变了对图像的表示方式而已,即将原来用 R、G、B 三个基色表示的图像信号,改用亮度/色度分量与其预测值之间的差值、运动矢量和空间频率分量等物理量来表示。在这些新的表示方式中,对图像质量影响大的信息集中到了少数的量上,以便于我们将这少数至关重要的量选择出来加以传送,达到降低数据率的目的。

量化是编码器中唯一产生信息损失的地方。通过量化,保留下重要的 DCT 系数,将其数值进一步离散化,并将不重要的 DCT 系数量化到零。这是对数据进行压缩的很重要的一步。经过 DCT 变换后,系数之间的相关性已经很小,因此可以用各自独立的量化器分别进行量化。对于 DC 分量,由于帧内编码相邻块的直流分量之间有很强的相关性,因而相邻块的 DC 分量在采用 DPCM 后(见 4.1.3 节),用 8 比特均匀(最小熵)量化器量化。其他系数也采用均匀量化器,但对不同的 DCT 系数根据其重要程度的不同采用不同大小的量化台阶。例如,低频系数的量化台阶较小,而高频系数的量化台阶则选得比较大。实现对不同系数使用不同量化台阶的最简单的方法是,将每个系数除以不同的权值,然后统一地采用同一个量化器进行量化。权值的大小由人的视觉对该频率分量的敏感程度确定。8×8 个权值构成一个矩阵,称为**量化矩阵**。

图 4-4(a) 和(b) 分别给出了帧内编码子块和帧间编码子块的量化矩阵的例子。对于只进行帧内编码的子块,DCT 系数值对应于该图像块的空间频率谱,因此,图(a) 所示量化矩阵中低频系数的权值较小,而高频系数的权值较大。而对于帧间编码的子块,由于帧间预测误差的 DCT 系数与原图像的空间频率谱关系很小,因而其直流和交流分量均采用相同的量化器量化。

8	16	19	22	26	27	29	34
16	16	22	24	27	29	34	37
19	22	26	27	29	34	34	38
22	22	26	27	29	34	37	40
22	26	27	29	32	35	40	48
26	27	29	32	35	40	48	58
26	27	29	34	38	46	56	69
27	29	35	38	46	56	69	83

(a)帧内编码

16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16
16	16	16	16	16	16	16	16

(b)帧间编码

图 4-4　量化矩阵

5. 从二维矩阵到一维序列的转换

DCT 系数矩阵在量化之后已经变得很稀疏，相当多的系数已量化到零，只剩下低频系数和少数的高频系数。为了进行熵编码，我们将按图 4-5(a)所示的之字形顺序，将二维的系数矩阵转化成一个一维的数据序列。图中所示的扫描顺序，首先经过低频再到高频。这样，重要的系数多数都集中在序列的开始。当扫描到高频区域遇到若干个零之后才有一个非零值时，为了记录该非零值的位置，该值前面的零的个数用**游程编码**表示。当剩下的所有系数均为零时，用符号 **EOB**(End Of Block)来代表序列的截止。以这种方法可以将 64 个系数有效地排列成一个短序列，而每一个系数则用一个数组(run，level)来表示，其中 run 表示系数在矩阵中的位置，level 表示它的值。例如，(3，−6)代表这个系数前面有 3 个零，它的值为−6。

0	1	5	6	14	15	27	28
2	4	7	13	16	26	29	42
3	8	12	17	25	30	41	43
9	11	18	24	31	40	44	53
10	19	23	32	39	45	52	54
20	22	33	38	46	51	55	60
21	34	37	47	50	56	59	61
35	36	48	49	57	58	62	63

(a)之字形扫描

0	4	6	20	22	36	38	52
1	5	7	21	23	37	39	53
2	8	19	24	34	40	50	54
3	9	18	25	35	41	51	55
10	17	26	30	42	46	56	60
11	16	27	31	43	47	57	61
12	15	28	32	44	48	58	62
13	14	29	33	45	49	59	63

(b)交替扫描

图 4-5　DCT 系数的扫描顺序

对隔行扫描的图像进行编码时，由于相邻扫描行来自不同的场，如果在一场时间内物体的位置有明显的移动，垂直方向上的相关性会降低，导致该方向空间频率分量的幅值下降较慢，之字形的顺序就不是最佳的了。这时可以采用图 4-5(b)所示的交替扫描方式，这种方式优先扫描了垂直方向的系数。

另一种处理办法是，为了增加垂直相关性，在做 DCT 之前，先取奇数场的 8 行，再取偶数场的 8 行，分别作 8×8 的 DCT，这时仍可采用图(a)方式进行扫描。在解码时，经过 IDCT 以后再恢复正常的行顺序。

6. 熵编码

已经在第 3 章中讲述过，熵编码是实现无损数据压缩的一种重要方法。可以采用霍夫曼编码，也可用采用算术编码来消除上面所形成的一维序列(包含 DCT 系数的位置和幅度信息)中的统计冗余信息，给序列中的每个符号指定一个码字，从而产生编码器输出的比特流。

从 3.10 节可知，无论是霍夫曼码还是算核编码，信源实际的概率模型必须与设计码表时所采用的模型相匹配，否则编码效率会显著降低。因此在一些国际标准中采用了自适应熵编码，即在编码过程中，根据图像相邻区域的已编码的数据，估计当前符号的概率分布，然后动态地更新码表；或者切换到符合当前分布的码表。在后一种情况下，编、解码器都有一组事先根据数据不同统计特性设计的码表。

在实际设计码表时，通常不是将一个 DCT 系数值看成一个符号，而是将游程编码后的组合(run，level)看成一个符号，给予一个码字，这样的码表有时称为**二维霍夫曼表**。在不同的国际标准中霍夫曼码表的具体形式有所不同。以 H.263/MPEG4 的 DCT 系数码表为例，该标准对出现频率较高的符号定义了 VLC 码表，而不常见的符号仍采用定长码表示。表 4-1 给出了它的 VLC 码表的前 20 个码字。该表是一个(last，run，level)的三维码表，last 代表是否是序列的最后一个符号(last＝1，“是”)，run 代表非零系数前面的零的个数，level 代表该系数的绝对值，系数的正、负则由码字的最后一个比特 s 表示，0 为正值，1 为负值。

表 4-1　VLC 码表(部分)

Last	Run	Level	码字
0	0	1	10s
0	1	1	110s
0	2	1	1110s
0	0	2	1111s
1	0	1	0111s
0	3	1	01101s
0	4	1	01100s
0	5	1	01011s
0	0	3	010101s
0	1	2	010100s
0	6	1	010011s
0	7	1	010010s
0	8	1	010001s
0	9	1	010000s
1	1	1	001111s
1	2	1	001110s
1	3	1	001101s
1	4	1	001100s
0	0	4	0010111s
0	10	1	0010110s
…	…	…	…

7. **去块效应滤波器**

从 3.9.5 节可知,块 DCT 直流及低频分量的量化误差会导致重建图像上的块效应。除此之外,在分块进行具有运动补偿的帧间预测时,相邻宏块的预测块通常来自于参考帧中的不同(非相邻)位置,它们在空间上的不连续性,是造成块效应的另一个原因。为了降低块效应引起的视觉不快,可以用一个低通滤波器来对解码后的重建图像进行平滑,此滤波器称为**去块效应滤波器**(Deblocking filter)。在图 4-1 中,虚线所示的去块效应滤波器放在了解码器的预测环路之中;为了不失配,编码器环路中也有一个同样的滤波器。这样做的好处是,在提高解码图像主观视觉质量的同时,降低了参考图像块效应(不连续性)产生的残差,有利于压缩码率的降低。

8. **速率控制**

由于编码的原因,例如熵编码产生的是变长码、帧内编码和帧间编码产生的数据量不同等,同时由于视频信号的统计特性不平稳,复杂度高的图像经压缩后所产生的数据量大,因此编码输出的比特流速率是变化的。为了在恒定比特率的信道上传送,需要在进入信道之前加一个缓冲器,用以对输出码流的速率进行平滑和控制。关于速率控制问题将在 4.3 节中做进一步的介绍。

在图 4-1 中还给出了解码器的功能方框图。解码是编码的逆过程,这里不再多述。

4.1.2　视频序列的编码

对一个视频图像序列进行压缩编码,需要用到帧间预测编码,因为它是消除时间冗余信息(即各帧图像之间的冗余信息)的一种有效方法,但在不同的应用中帧间预测的具体做法有所不同。由于后向或双向预测在解码显示时引入额外的延时,所以在实时性要求高的应用中,如可视电话和视频会议,一般不采用。在这些应用中,通常第一帧采用帧内编码,后继帧均采用前向预测编码。

在另外一些应用中,例如在数字电视广播中接收机刚开机或改换频道时,或在视频点播中接收机进行录像机式的快进、快退等操作时,需要从码流的某个中间点(不是序列的起始点)处开始解码,这称为**随机接入**(Random Access)。如果序列除第 1 帧外全部采用前向预测,此时会因为没有前一帧参考图像而不能解码。为了解决随机接入的问题,有必要隔一段时间传送一帧只采用帧内编码的图像作为参考图像,使得预测环路能在没有某一个参考帧时,迅速获得另一个参考帧,从而恢复到正常工作状态。图 4-6(a)表示出这样一个图像序列,图中 I 帧表示帧内编码图像,P 帧和 B 帧分别表示采用前向预测和双向预测的图像。P 帧由与其相邻的前一个 P 帧或 I 帧来预测,B 帧则由前、后相邻的 I 帧或 P 帧预测。在这类应用中,实时性要求不那么苛刻,因此可以使用 B 帧提高压缩比,但 B 帧一般不作为 P 帧和后续 B 帧的参考图像。两个 I 帧之间的图像构成一个组,称为 **GOP**(Group Of Picture)。在一组图像中,P 帧和 B 帧的数目可以根据实际应用的需要选择。

需要注意,在 GOP 中 B 帧既引起编码的延时,也引起解码的延时。如图 4-6(a)所示的情况,只有在 P_4 编码之后才能对 B_2 进行编码,与输入序列相比,编码器引入了 2 帧(连续出现的 B 帧的个数)的延时。为了使解码器引入的延时最小,已编码的图像按如下顺序进行存储或传输:I_1 P_4 B_2 B_3 P_8 B_5 B_6 B_7…不难看出,要正确地解码和显示该序列图像,解码器需要存储 2 帧已解码的图像,

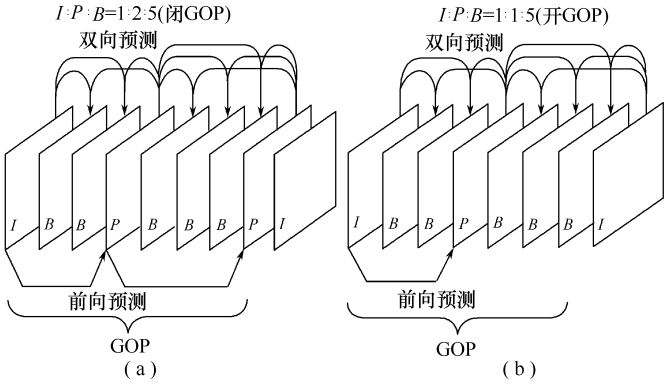


图 4-6　使用 I 帧、P 帧和 B 帧编码的图像序列

而引入的延时为 1 帧。

当 GOP 以 B 帧为结束帧时[见图 4-6(b)]，最后的 B 帧的编、解码需要以下一个 GOP 的 I 帧作为参考帧，这样的 GOP 称为**开 GOP**；当 GOP 以 P 帧为结束帧时[见图 4-6(a)]，不需要参考下一个 GOP 的 I 帧，这样的 GOP 称为**闭 GOP**。

4.1.3　帧内预测编码

本节介绍基于空间域预测编码的优化技术。

1. DC 系数的帧内预测编码

每个 8×8 帧内编码块 DCT 系数中的 DC 分量代表了该块的平均灰度，而在空间域内图像的灰度是连续变化的，因此相邻块的 DC 分量之间存在相关性。利用空间邻域已编码块的 DC 分量对当前块的 DC 分量进行预测，然后对预测误差进行编码，可以降低 DC 系数所占用的比特数。

图 4-7(a)给出了进行 DC 系数预测的相邻块与当前块的位置图。当前块 X 的 DC 系数可以直接由 A 块的 DC 系数预测，例如在 MPEG－2 中就采用了这种方法。更好的方式(如 MPEG－4 所采用)是按下式选择 DC 系数梯度变化较小的方向进行预测。

$$\text{If } |DC_A - DC_B| < |DC_B - DC_C|$$

从 C 预测 X

else

从 A 预测 X

一般取 A 或 C 的 DC 系数乘以一个小于 1 的权值作为块 X 的 DC 系数的预测值。

同理，空间相邻的帧内编码块的 DCT 系数的低频分量之间也存在着一定的相关性，因此，DCT 系数矩阵的第一行和第一列也可以利用帧内预测编码来降低它们占用的比特数。图 4-7(b)给出了低频 AC 系数(图中灰色位置)的预测关系。

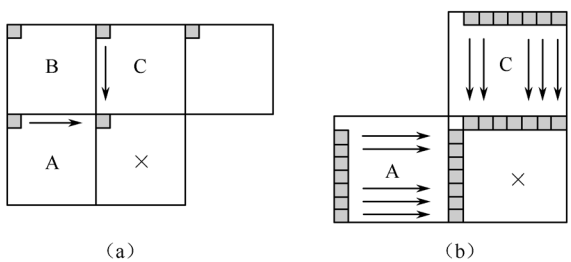


图 4-7　(a)DC 系数的预测，(b)AC 系数的预测

2. 运动矢量的帧内预测编码

当图像中存在面积覆盖若干宏块的大物体时，空间相邻宏块的运动矢量之间存在着相关性，因此，对运动矢量采用帧内预测编码有利于降低传送运动矢量所需要的比特数。

图 4-8 给出当前块 E 和与其相邻块 A、B 和 C 的相对位置，其中(a)为进行运动估值的块大小相同(如都是 16×16)时的情况。在有些国际标准中允许使用不同大小的块来进行运动估值(见 4.1.4 节)，图(b)表示出 E 和邻域块大小不同时的一个例子。最简单的预测方法是用左邻域块的运动矢量 MV_A 作为当前块 E 的运动矢量 MV_E 的预测值；更好的方法如下式所示：

$$MV_{Ex}=\text{median}\{MV_{Ax},MV_{Bx},MV_{Cx}\}$$
$$MV_{Ey}=\text{median}\{MV_{Ay},MV_{By},MV_{Cy}\} \tag{4-1}$$

式中，下标 x、y 分别代表运动矢量在 X 和 Y 方向上的分量，median{·}代表取中值。如果当前块是 16×8 或 8×16 的，则分别按图(c)和(d)中箭头所示方向进行预测。最后，对当前运动矢量与预测值之差进行编码和传输。

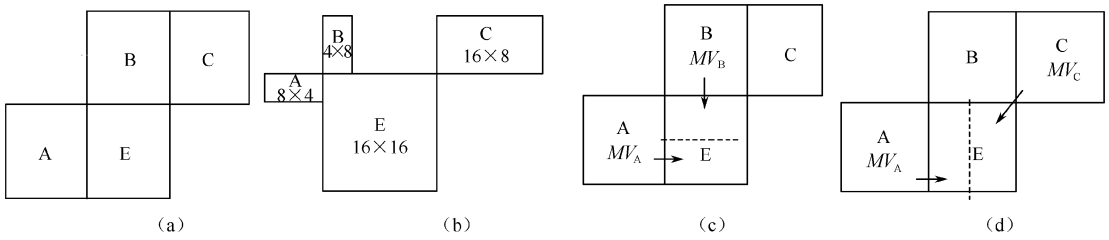


图 4-8　运动矢量预测中当前块与邻块的位置关系

3. 图像的帧内预测编码

在 4.1.1 节中讲过，对于帧内编码的帧(或块)不做帧间预测，直接进行块 DCT 和熵编码。但实际上，图像在空域内的相关性是很强的，如果对于帧内编码模式，首先进行空间域预测编码，再作 DCT 和熵编码可以明显的提高编码效率。因此在某些国际标准(如 H.264)中，增加了帧内预测编码的模式。

根据 3.5.1 节可知，预测器系数的设计与图像信号的内容(统计特性)有关。为了适应不同的图像内容，通常给出一组不同的预测器。图 4-9 列出了几个例子。图中浅灰色块表示当前待预测像素，大写字母表示邻接的像素。图(a)定义了各像素的相对位置；图(b)用顶部像素分别预测当前块对应列的各像素；图(c)用 A、D、I、L 的均值预测当前块各像素；图(d)中像素由周边像素的加权和来预测，例如

$$d=\text{round}\left(\frac{B}{4}+\frac{C}{2}+\frac{D}{4}\right) \tag{4-2}$$

$$c=h=\text{round}\left(\frac{A}{4}+\frac{B}{2}+\frac{C}{4}\right) \tag{4-3}$$

显然，图(b)和图(d)分别适用于垂直方向和对角线方向相关性强的图像，而图(c)适用于灰度变化平坦的图像。类似地，相关的国际标准(如 H.264)中还规定了其他一些预测器。

4.1.4　帧间预测编码的优化

在图 4-1 所示的混合编码器中，帧间预测编码对数据压缩起着至关重要的作用，因此对其进行改进，一直是近年来视频压缩编码研究的一个重要组成部分。下面简单介绍这些研究所获得的部分成果，它们已分别在不同的国际标准中被采用。

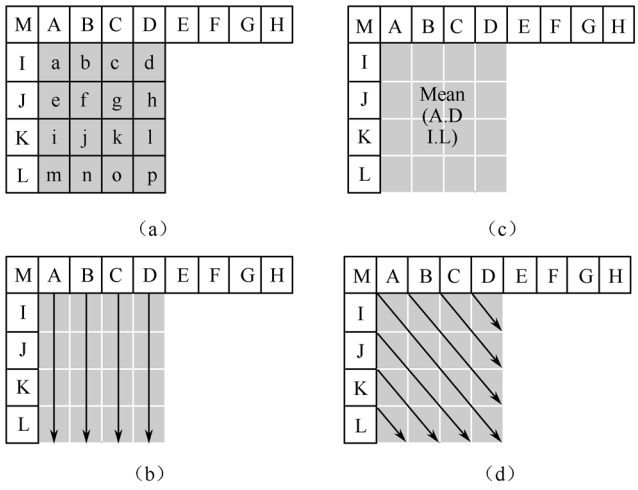


图 4-9　亮度图像帧内预测的几个模式

1. 隔行扫描图像的预测

当编码器的输入是隔行扫描的图像时，由于相邻扫描行来自不同的场，如果在一场时间内物体的位置有明显的移动，垂直方向上的相关性会降低。在这样的情况下，为了提高预测的准确性，可以采用以下的模式：

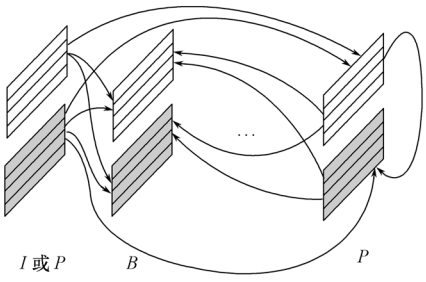


图 4-10　场图像的场间预测

（1）用于场图像的场间预测

这种方式与帧间预测类似，只是被预测的宏块是从某一场图像（而不是帧图像）中取出的。如图 4-10所示，P 帧某一场的参考图像可以选择为该场图像的前一场，或者与之极性（即奇数场或偶数场）相同的前一场图像；而 B 帧某一场的参考图像则可以由其前一帧的任何一场与后一帧的任何一场构成。

（2）用于帧图像的场间预测

将帧图像中的宏块分成两个场宏块，两个场宏块分别由奇数场和偶数场的像素构成。对这两个场宏块分别进行场间运动估值。这样，对前向预测而言，每个帧宏块有两个运动矢量；对双向预测而言，每个帧宏块则有 4 个运动矢量。这种预测方式主要用于图像运动剧烈的情况。

2. 帧间预测编码中几种特殊情况的处理

（1）不受限的运动矢量

对靠近图像边缘处的宏块进行运动估值时，其匹配块可能部分地落在参考图像之外（见图 4-11）。为了降低预测误差，从而降低此时编码所需的比特数，可以将图像边界上的像素值简单线性外插到边界外（如图中阴影所示），编码效率会因此而略有提高。

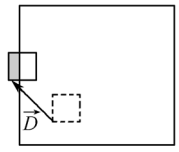


图 4-11　匹配块部分地落在参考图像之外的例子

（2）跳过模式

若当前编码块以参考图像中的同位块作为预测块时，其预测误差足够小（可以量化到零），则不传送当前块的任何编码信息，并称其为**跳过（skip）块**。在解码时，直接取参考图像中同位块作为跳过块的解码值。

（3）双向直接模式

在这种模式中，只传送预测误差，而不传送当前编码块的运动矢

量，当前块的两个双向预测运动矢量通过已编码块的运动矢量推导出来。以图 4-12 所示的情况为例。设当前编码块位于 B_5 帧，其后向参考帧 P_6 中的同位块的运动矢量指向 I_1，且数值为 $MV=(+2.5,+5)$。那么当前块的前向参考帧也为 I_1，其运动矢量按下式计算：

$$\text{前向预测 } MV_F = \frac{B_5 \text{ 到 } I_1 \text{ 的帧数}}{P_6 \text{ 到 } I_1 \text{ 的帧数}} \cdot MV = (2,4) \tag{4-4}$$

$$\text{后向预测 } MV_B = \frac{B_5 \text{ 到 } P_6 \text{ 的帧数}}{P_6 \text{ 到 } I_1 \text{ 的帧数}} \cdot MV = (-0.5,-1) \tag{4-5}$$

当前块的解码值则等于由 MV_F 和 MV_B 决定的双向预测值与预测误差之和。

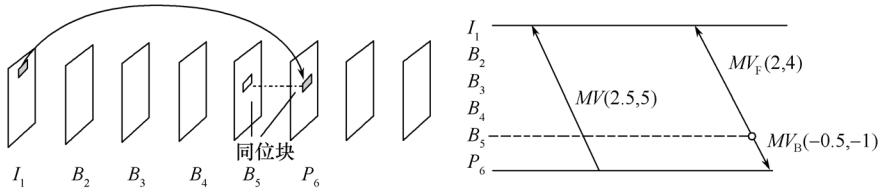


图 4-12　双向预测中的直接模式

3. 运动估值的图像块大小

如 3.5.2 节所述，在块匹配方法中，将图像分块并将每一个块视为一个“运动物体”，对于图像结构和运动来说，这是一个相当粗糙的模型。如果一个块中包含若干个向不同方向运动的物体，则很难在参考帧中找到合适的匹配块。

根据图像内容的复杂程度选择不同大小和形状的块来进行运动估值，可以在一定程度上改善上述状况，减小预测误差，从而降低码率。因而在一些国际标准中除了 16×16 的块以外，还允许在运动估值中采用其他的分块方式。图 4-13 给出了 H.264 支持的各种分块模式，其中包括 16×16、16×8、8×16 和 8×8 的块；当选择 8×8 的块时，可以进一步细分为 8×4，4×8 和 4×4，这常称为**树结构**的层次化块划分。值得注意的是，更细地划分块在可能降低预测误差的同时，增加了需要传输的运动矢量的个数（例如对 4×4 的前向预测，每个宏块有 16 个运动矢量）和匹配搜索的复杂程度；同时，所选择的是多大的块（**块模式**）也必须作为附加信息传送给解码器。

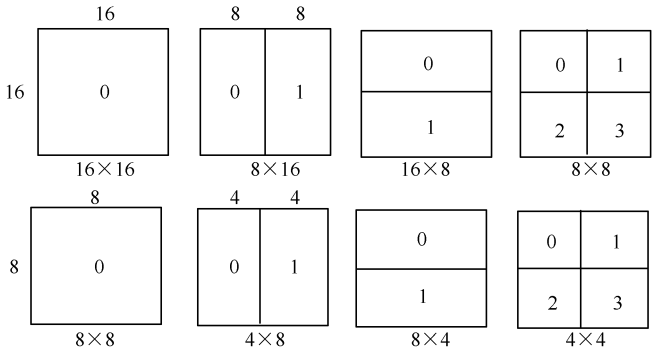


图 4-13　分层划分的块模式

对一个图像宏块进行运动估值时，究竟划分为多大的块好呢？原则上说，需要对所有可划分的模式进行试探，看哪一种模式下所需的编码（预测误差、运动矢量和块模式等的编码）总比特数最少。由于图 4-13 所示的 7 种模式都在一个宏块之中，相互之间有很强的相关性，如何利用这些相关性就成为研究可变块大小的运动估值快速算法的基本出发点。一种思路是自下而上，先对最小块（4×4）进行运动估值，然后对小块的运动矢量进行分析，例如如果 4 个4×4块的运动矢量相似，则可合并为 8×8 的块；同时，4×4 块的 SAD 值可以在进行大块的搜索时再利用。第二种思路是

自上而下，先对最大块(16×16)进行运动估值，然后用大块的运动矢量对小块的位移进行预测，以减少搜索点的数目。考虑到太小的块容易产生误匹配，所得到的运动矢量不够可靠；而太大的块又容易包含多个向不同方向运动的物体，致使利用大块的运动矢量对小块的运动矢量进行预测不够准确，因此一些算法以 8×8 块为初始搜索块，然后采取向上合并和向下分裂的策略，以得到最佳的分块模式。4.1.5 节将对进行模式选择的方法做更深入的讨论。

4. 亚像素精度运动矢量估计

在 3.5.2 节所介绍的运动估值中，当前帧和参考帧具有相同的空间分辨率，这样得到的当前块和它在参考帧中的匹配块之间的位移值是像素取样间隔的整数倍，或者说，所得到的运动矢量是整数像素精度的。很显然，物体的实际帧间位移与像素点的取样间隔之间并没有必然的联系。对实际位移进行更精细(亚像素精度)的估值，有可能进一步减小帧间预测误差从而降低码率。由于在进行亚像素(例如 1/2 像素、1/4 像素等)精度的运动估值时，首先需要通过内插得到参考图像在亚像素位置上的数值，以便实现对当前块位移的更精细的匹配，因此对亚像素精度运动估值的研究，包括运动矢量的精度和参考图像的内插方法分别对降低预测误差的影响这样两个问题。

(1) 运动矢量的精度对降低预测误差的影响

研究表明[1]，运动矢量从整像素精度提高到亚像素时，帧间预测误差有显著降低，但到了一定的“临界精度”之后，预测误差降低的可能性则减小。对一系列典型测试序列的实验表明[1]，当运动矢量的精度从整像素提高至 1/8 像素时，编码比特率的降低比较明显；当提高至 1/16 像素后，则无明显的进一步的改善。同时，具有小物体、或低信噪比的序列相对于大物体、或高信噪比的序列而言，提高运动估值精度的效果相对要差，这是因为如果一个块中包含向不同方向运动的物体，或者量化误差较大，都会对运动估值形成干扰，进一步提高精度则没有多大意义。

(2) 内插滤波器对降低预测误差的影响

从概念上讲，只要如 3.4.2 节所述采用理想低通作为内插滤波器，就可以精确地恢复出参考图像在亚像素位置上的数值。但是实验表明，按照上述方法进行具有运动补偿的帧间预测，效果并不理想。这是因为输入信号中包含着摄像机噪声和其他信号成分，这些噪声不能被很好地预测从而保留在预测误差之中。研究者通过假设噪声的主要成分来源于输入信号中的混叠失真[2]，从理论和实验上比较好地解释了上述问题。

为了分析的简单，我们只考虑一维的情况。假设 $t-1$ 和 t 时刻的输入模拟图像信号分别为 $l_{t-1}(x)$ 和 $l_t(x)$，二者分别取样后得到 $b_{t-1}(x_n)$ 和 $b_t(x_n)$。若 $l_{t-1}(x)$ 的频谱为 $L_{t-1}(\omega)$，则 $b_{t-1}(x_n)$ 的频谱为

$$B_{t-1}(\omega)=\frac{1}{T}\sum_{k=-\infty}^{\infty}L_{t-1}(\omega-k\omega_s) \tag{4-6}$$

式中，ω_s 为取样角频率；T 为取样周期。不失一般性，令 $T=1$，以忽略上式中的常数 $1/T$。

假设在数据采集过程中，由于取样之前的限带滤波器不是一个理想的低通，没有在 1/2 的取样频率(但在 1 倍的取样频率)上将信号衰减到零，即

$$\begin{aligned} L_{t-1}(\omega)&\neq 0 \qquad |\omega|\geqslant\omega_s/2\\ &=0 \qquad |\omega|\geqslant\omega_s \end{aligned} \tag{4-7}$$

则取样后的离散信号 $b_{t-1}(x_n)$ 的频谱 $B_{t-1}(\omega)$ 中存在混叠干扰。让我们只考虑该频谱的基带部分(见图 4-14)，

$$B_{t-1}(\omega)=L_{t-1}(\omega)+L_{t-1}(\omega+\omega_s)+L_{t-1}(\omega-\omega_s) \tag{4-8}$$

假设 $l_t(x)$ 只是 $l_{t-1}(x)$ 平移了 d_x 后的结果，即 $l_t(x)=l_{t-1}(x-d_x)$，那么取样后的当前帧信号 $b_t(x_n)$ 的频谱为

$$B_t(\omega)=l_{t-1}(\omega)\mathrm{e}^{-\mathrm{j}d_x\omega}+L_{t-1}(\omega+\omega_s)\mathrm{e}^{-\mathrm{j}d_x(\omega+\omega_s)}+L_{t-1}(\omega-\omega_s)\mathrm{e}^{-\mathrm{j}d_x(\omega-\omega_s)} \tag{4-9}$$

上式后两项代表 $B_t(\omega)$ 中的混叠成分。根据(4-7)式可知，这两个混叠项是互不重叠的(见图 4-14)，因此(4-9)式可化为

$$B_t(\omega) = l_{t-1}(\omega)\mathrm{e}^{-\mathrm{j}d_x\omega} + L_{t-1}\left[\omega\left(1-\frac{\omega_s}{|\omega|}\right)\right]\mathrm{e}^{-\mathrm{j}d_x\left(1-\frac{\omega_s}{|\omega|}\right)\omega} \tag{4-10}$$

由(4-10)式可以看出，在基带内混叠项的位移 d_{xa} 为

$$d_{xa} = -d_x\left(\frac{\omega_s}{|\omega|}-1\right) \qquad -\frac{\omega_s}{2} < \omega < \frac{\omega_s}{2} \tag{4-11}$$

(4-11)式表示混叠项位移 d_{xa} 的大小与频率 ω 有关，同时 d_{xa} 与图像本身的位移是相反方向的。由于运动补偿只能按照一个位移矢量进行，在对图像本身的运动进行补偿时，混叠项将遗留在预测误差之中，造成补偿效果的不理想。

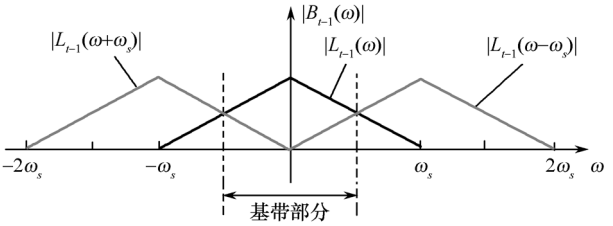


图 4-14　输入信号 $b_{t-1}(x_n)$ 的频谱

下面我们来具体分析混叠失真对预测误差的影响。图 4-15 给出了具有运动补偿的预测器的一般框图。图中 $\hat{b}_{t-1}(x_n)$ 和 $\tilde{b}_t(x_n)$ 分别代表重建的前一帧参考图像和对当前帧的预测值，$l'_{t-1}(x)$ 代表经内插滤波器 $h(x)$ 得到的模拟参考图像，$\hat{l}_t(x) = l'_{t-1}(x-\hat{d}_x)$，$\hat{d}_x$ 为运动矢量的估计值。在这里我们假设其估计值等于实际值，即 $\hat{d}_x = d_x$。最后一个模块对模拟信号 $\hat{l}_t(x)$ 以周期 T=1 进行再取样。

根据图 4-15 很容易得到频域内各信号间的关系式：

$$L'_{t-1}(\omega) = H(\omega)\hat{B}_{t-1}(\omega) \tag{4-12}$$

$$\hat{L}_t(\omega) = L'_{t-1}(\omega)\mathrm{e}^{-\mathrm{j}d_x\omega} \tag{4-13}$$

$$\tilde{B}_t(\omega) = \sum_{k=-\infty}^{\infty}\hat{L}_t(\omega-k\omega_s) \tag{4-14}$$

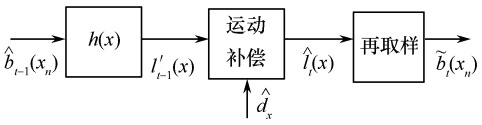


图 4-15　具有运动补偿的帧间预测器一般框图

式中，$H(\omega)$ 为内插滤波器的频率响应。如果内插滤波器是理想低通型，即

$$H(\omega) = \begin{cases} 1 & |\omega| < \omega_s/2 \\ 0 & \text{其他} \end{cases} \tag{4-15}$$

假设忽略量化误差，即参考图像等于原图像，那么根据(4-12)式和(4-15)式，$L'_{t-1}(\omega) = B_{t-1}(\omega)$，且在 $|\omega| > \omega_s/2$ 时，$L'_{t-1}(\omega) = 0$。因此根据(4-8)式和(4-13)式，在基带中有：

$$\tilde{B}_t(\omega) = L_{t-1}(\omega)\mathrm{e}^{-\mathrm{j}d_x\omega} + L_{t-1}(\omega+\omega_s)\mathrm{e}^{-\mathrm{j}d_x\omega} + L_{t-1}(\omega-\omega_s)\mathrm{e}^{-\mathrm{j}d_x\omega} \tag{4-16}$$

根据(4-10)式和(4-16)式得到预测误差的频谱 $E(\omega)$ 为

$$E(\omega) = B_t(\omega) - \tilde{B}_t(\omega) = L_{t-1}\left[\omega\left(1-\frac{\omega_s}{|\omega|}\right)\right]\left[\mathrm{e}^{-\mathrm{j}d_x\left(1-\frac{\omega_s}{|\omega|}\right)\omega} - \mathrm{e}^{-\mathrm{j}d_x\omega}\right]$$

$$= 2L_{t-1}\left[\omega\left(1-\frac{\omega_s}{|\omega|}\right)\right]\cdot\sin\left(\frac{d_x\omega_s\omega}{2|\omega|}\right)\cdot\mathrm{e}^{-\mathrm{j}d_x\left(\omega-\frac{1}{2}\omega_s\omega/|\omega|\right)+\mathrm{j}\pi/2} \tag{4-17}$$

上式表示在对图像进行理想的预测时，对混叠项的预测是不完善的。从(4-17)式看出，残留在预测误差中的混叠失真大小(绝对值)与位移量 d_x 有关，在整像素位移($d_x = 0, \pm T, \pm 2T, \cdots$)时，残留误差消失，而在半像素位移时最大。

为了减小预测误差中残存的混叠失真，我们必须修正预测信号((4-16)式)，使其更接近于当前帧信号 $B_t(\omega)$ [(4-10)式]。由于亚像素位移产生混叠残留误差，故而进行内插的滤波器 $h(x)$ (见图 4-15)可以兼用来对预测信号中的混叠失真进行衰减。我们知道在最小均方误差准则下，对带有加性噪声的随机信号进行预测的最佳预测器是**维纳**(Wiener)**滤波器**，因此在国际标准中建议的内插滤波器是根据信号和噪声统计特性设计的维纳滤波器，它具有与一般低通类似的频谱形状，不同的是它考虑了混叠失真(噪声)的影响。实际上，(3-33)式就是 H.264 给出的一个 6 抽头的维纳滤波器；而双线性内插滤波器等效于长度为 2 的维纳滤波器。

5. 多参考帧预测

在对当前帧进行帧间编码时，通常采用前(或后)一帧作为它的参考图像，但在某些情况下，采用前面(或后面)若干帧中的某一帧作为参考可能会获得更好的预测效果(更低的预测误差)。这些情况包括：

(1) 被遮挡和新暴露出的背景——背景被运动物体或阴影遮挡，当它重新暴露出来时，可能在它的前一(被遮挡)帧中找不到匹配部分，而在数帧之前它未被遮挡的帧中，可以得到良好的匹配；

(2) 物体重复性的运动，或摄像机抖动、角度交替交换等，在这些情况下当前帧可能与数帧之前(或之后)的图像更具相似性；

(3) 由于运动相对于取样栅格的变化、照明变化、噪声或其他原因，尽管在视觉上并不一定感觉数帧前(或后)的图像与当前帧更相像，但对某些宏块而言，却往往可以在数帧前(或后)的图像中找到比前一帧为参考的更好的匹配。

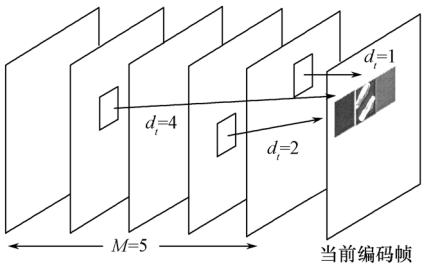


图 4-16　多参考帧帧间预测

由此可见，采用多参考帧预测能够在一定程度上提高编码效率。图 4-16 给出了一个采用多参考帧预测的情况，当前帧中左边黑色块的参考块在前 2 帧中，而中间条纹块的参考块则在前面第 4 帧中(对于反向和双向预测亦类似)。此时，对于每个宏块而言，不仅要传送预测误差和运动矢量，还要传送其参考块所在的图像的位置。多参考帧帧间预测编码效率的提高是以加大运算量(需要在多帧中搜索匹配块)和存储容量(需要存储多个参考帧)为代价的，同时，编、解码器需要同步地管理参考帧缓存区也增加了复杂程度，但是并不加大播放延时。

4.1.5　基于率－失真优化的编码模式选择

在视频编码标准中，通常允许进行多种模式的编码，例如 H.264 支持前面介绍的各种模式的编码，即一个宏块在编码的时候，可以采用帧内编码或帧间编码，而帧内编码有图 4-9 举例的多种预测模式，帧间编码也可以采用图 4-13 所示的多种预测模式，那么究竟采用哪种模式好呢？最简单和原始的判断准则是最小化预测误差，如在 3.5.3 节中根据所给出的 DFD 值最小来选择前向、后向或双向预测模式。根据经验，预测误差小所需的编码比特数就少，但是上述准则并没有全面考虑一个宏块编码所需的总比特数。除了预测误差之外，运动矢量和**编码模式**(如帧内/帧间、块大小和多帧参考中的参考帧标号等)也需要占用一定的比特，在低码率的应用中，这部分开销尤为不可忽视。

根据 3.3 节介绍的率－失真理论，一个更为合理的编码模式选择准则是率－失真优化(Rate Distortion Optimization)，简称 **RDO**。假设信源样值(或宏块)集合(序列)为 $\boldsymbol{S}=(S_1,\cdots,S_K)$，$I_k$ 为编码 S_k 所用的模式，且 $I_k\in\boldsymbol{O}$，$\boldsymbol{O}$ 为所有可选择的编码模式(包括量化器)的集合，那么在不考虑噪声的情况下，用于该信源序列的编码模式应该满足

$$\min_{\boldsymbol{I}} \quad D(\boldsymbol{S},\boldsymbol{I}) \tag{4-18}$$
$$\text{Subject to} \quad R(\boldsymbol{S},\boldsymbol{I}) \leqslant R_C$$

其中 $\boldsymbol{I}=(I_1,\cdots,I_K)$，$D(\boldsymbol{S},\boldsymbol{I})$ 和 $R(\boldsymbol{S},\boldsymbol{I})$ 分别为选择编码组合 $\boldsymbol{I}$ 时的总失真和总速率，R_C 为给定的速率限制值。引入拉格朗日参数 $\lambda \geqslant 0$，(4-18)式的条件优化问题变为

$$\min_{\boldsymbol{I}} J(\boldsymbol{S},\boldsymbol{I}) = \min_{\boldsymbol{I}} [D(\boldsymbol{S},\boldsymbol{I}) + \lambda R(\boldsymbol{S},\boldsymbol{I})] \tag{4-19}$$

其中 $J(\boldsymbol{S},\boldsymbol{I})$ 称为**代价函数**。

理论上讲，$D(\boldsymbol{S},\boldsymbol{I})$ 应该等于原始图像与解码重建图像的差值，$R(\boldsymbol{S},\boldsymbol{I})$ 包括量化后的预测误差(对帧内模式则是图像本身)的 DCT 系数、运动矢量和所选模式的标识符等编码所用的比特数之和。显然，(4-19)式是一个复杂的优化问题，再考虑到混合编码中，许多编码参数是经过空域或时域预测以其差值传送的，一个宏块参数的编码涉及到邻域或邻帧的其他宏块，这就使得问题更加复杂化。为此，研究者提出了多种简化方案来减少优化的搜索空间，以适应实际编码器设计的需要。

一个简单和被广泛接受的方案是假设失真和速率度量均为加性的，且仅依赖于每个宏块本身的编码参数的选择，则(4-19)式化为

$$\min_{\boldsymbol{I}} \sum_{k=1}^{K} J(S_k,\boldsymbol{I}) = \sum_{k=1}^{K} \min_{I_k} J(S_k,I_k) \tag{4-20}$$

式(4-20)中 K 为一帧宏块的个数，求和与求最小符号可以互换是因为对所有 k，$J(S_k,\boldsymbol{I})>0$。从上式右端可知，优化问题可以针对每个宏块独立进行。因此对于每个宏块我们有：

$$\min_{I_k \in \boldsymbol{O}} J(S_k,I_k/Q) = \min_{I_k \in \boldsymbol{O}} [D_{REC}(S_k,I_k/Q) + \lambda_{MODE} R_{REC}(S_k,I_k/Q)] \tag{4-21}$$

其中 Q 代表给定的量化步长，$\boldsymbol{O}$ 在不同标准中代表不同的集合，例如对于逐行扫描的 P 帧而言，在 MPEG 2 中，$\boldsymbol{O}$={Intra，Skip，Inter_16×16}；在 H.263/MPEG 4 中，$\boldsymbol{O}$={Intra，Skip，Inter_16×16，Inter_8×8}；在 H.264 中，$\boldsymbol{O}$={Intra_4×4，Intra_16×16，Skip，Inter_16×16，Inter_16×8，Inter_8×16，Inter_8×8，Inter_8×4，Inter_4×8，Inter_4×4}。上式中的失真项

$$D_{\text{REC}}(S_k,I_k/Q) = \sum_{(x,y)\in A_k} |S(x,y,t) - S'(x,y,t)|^2 \tag{4-22}$$

式中，A_k 为当前宏块。$S'(x,y,t)$，对 Intra 模式而言，是由经变换和量化的系数重建得到的像素值；对 Skip 模式，是前一帧同位块的重建值；而对 Inter 模式，是对经变换和量化的预测误差进行重建后的像素值。(4-21)式中的速率项 R_{REC}，对 Intra 模式而言，是宏块经熵编码后使用的比特数；对 Skip 模式，为 1 个比特/宏块；而对 Inter 模式，则为模式标识、参考帧标识、运动矢量和预测误差所使用的总比特数。图 4-17 表示出计算(4-21)式的**率－失真代价**(RD-cost)的流程。

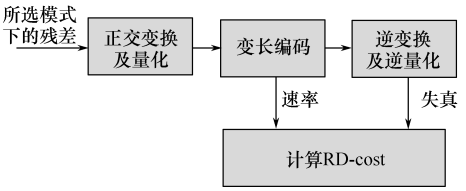


图 4-17　率－失真代价计算

由于帧间预测模式涉及运动补偿，因此我们将这一部分分成两个串行的优化问题来解决。对每一种帧间预测模式，首先忽略对预测误差进行编码(变换、量化和熵编码)的过程，只对帧间编码的各种模式进行速率约束下的运动估值，即找到最佳的运动矢量 $\boldsymbol{v}_{\text{opt}}$ 使下列代价函数最小：

$$\min_{\boldsymbol{v}\in \boldsymbol{V}} [D_{\text{DFD}}(S_k,\boldsymbol{v}) + \lambda_{\text{MOTION}} R_{\text{MOTION}}(S_k,\boldsymbol{v})] \tag{4-23}$$

其中 $\boldsymbol{V}$ 代表所有可能的运动矢量的集合，$R_{\text{MOTION}}(S_k,\boldsymbol{v})$ 代表采用 $\boldsymbol{v}$ 时，运动矢量编码所用的比特数；在多帧预测中，$\boldsymbol{v}=(v_x,v_y,v_t)$，其中 v_t 代表参考帧的标识。上式中的失真项为

$$D_{\text{DFD}}(S_k,\boldsymbol{v}) = \sum_{(x,y)\in A_k} |S(x,y,t) - S'(x-v_x,y-v_y,t-v_t)|^p \tag{4-24}$$

其中 $S(x,y,t)$ 和 $S'(x,y,t)$ 分别为原始和预测的像素值。当 $p=1$ 时，(4-24)式为 SAD；$p=2$ 时，

为平方误差和(SSD)。(4-23)式首先在整像素精度上求解,找到最佳的整像素精度运动矢量后,再在该像素周围的亚像素精度上进行搜索,直至找到满足(4-23)式的最佳运动矢量 $\boldsymbol{v}_{\text{opt}}$。然后,按照图 4-17 所示的过程计算率－失真代价,与其他帧间预测模式和帧内模式一起进行模式选择。

在具体实现上述优化时,我们仍有如下问题需要解决:

(1) 拉格朗日参数的选择

假设率—失真函数 $D_{\text{REC}}(R_{\text{REC}})$ 是凸的,那么(4-21)式的代价函数 $J_{\text{MODE}}(R_{\text{REC}}) = D_{\text{REC}}(R_{\text{REC}}) + \lambda_{\text{MODE}} R_{\text{REC}}$ 也是凸的。令 J_{MODE} 的一阶导数为零,则可得到 J 取最小值时

$$\lambda_{\text{MODE}} = -\frac{\mathrm{d}D_{\text{REC}}}{\mathrm{d}R_{\text{REC}}} \tag{4-25}$$

上式说明拉格朗日参数 λ_{MODE} 对应于率－失真函数的负斜率,因此可以从率－失真函数来确定 λ。在假设足够高码率的条件下,可以得到熵约束量化器的率－失真函数的近似表达式和失真与 Q 之间的关系式,并通过实验确定其中的常数,从而得到如下的拉格朗日参数[3]。

$$\begin{aligned} &\text{对 H.263/MPEG 4:} && \lambda_{\text{MODE}} = 0.85QP^2 \\ &\text{对 H.264/AVC:} && \lambda_{\text{MODE}} = 0.85 \times 2^{(QP-12)/3} \end{aligned} \tag{4-26}$$

其中 QP 为编码**量化参数**,在不同的标准中,QP 与 Q 之间有不同的关系(例如表 4-2)。在(4-24)式中,若 DFD 取 SAD 值,则 $\lambda_{\text{MOTION}} = \sqrt{\lambda_{\text{MODE}}}$;若取 SSD 值,则 $\lambda_{\text{MOTION}} = \lambda_{\text{MODE}}$。考虑到 D_{REC} 和 R_{REC} 都与量化步长 Q 有关,λ_{MODE} 与 QP 有(4-26)式的依赖关系就是可以理解的了。

(2) 快速模式选择算法

在进行(4-23)式和(4-21)式的优化时,运动矢量编码所需的比特数 R_{MOTION} 可以通过 VLC 表迅速查出,但是要得到帧间或帧内预测误差经变换、量化(对所有可能的量化步长)和变长编码及运动矢量等所占用的比特数 R_{REC},以及重建后产生的失真 D_{REC},则需要针对每一种模式完成多次(多个 QP)编码/解码的全过程,计算量是十分庞大的。因此,从基于率－失真优化的编码模式选择方法提出以来,对其快速算法的研究就一直是一个活跃的领域。这些快速算法的基本思想通常是:①根据图像在空间域与时间域上的特征或相关性和邻域已编码模块的信息,限定搜索模式的种类,或确定优先搜索的模式(或模式类别);②设定适当的阈值,当代价函数小于阈值时即认为达到最优,使搜索及早中止;或当代价函数大于某个阈值时,及早跳转到对其他模式的搜索。对快速算法有兴趣的读者可以参阅[4]～[7]。

(3) 根据预测误差对失真和速率进行估计

这也是简化模式选择计算复杂度的一种方法。例如利用一个大的训练数据集,得到以量化参数 QP 为参变量的 $D_{\text{REC}} \sim DFD$ 和 $R_{\text{REC}} \sim DFD$ 近似关系。这些关系可以表格的方式存储,在进行(4-21)式的优化时,通过查表而不必完成编解码的全过程就可以获得对应的失真和速率数值。

4.1.6 低计算复杂度的变换编码与量化

8×8 DCT 是大多数视频编码国际标准中采用的变换编码方式,但是 DCT 通过浮点计算完成,由于有限字长效应,且编码器和解码器的计算单元的精度可能不同,因此逆变换不一定能精确地重建变换前的数值,这称为编、解码变换的**失配**。失配造成解码图像质量一定程度上的降低。为了解决变换失配问题并降低变换的计算复杂度,在 H.264 中引入了一种 4×4 的正交变换。

1. 4×4 正交变换

我们首先从 4×4 DCT 开始来推导这种低计算复杂度的 4×4 变换。根据(3-67)式得到以矩阵形式表示的 4×4 DCT 如下式所示:

$$\boldsymbol{Y}=\boldsymbol{A}\boldsymbol{X}\boldsymbol{A}^{\mathrm{T}}=\begin{bmatrix}a & a & a & a\\ b & c & -c & -b\\ a & -a & -a & a\\ c & -b & b & -c\end{bmatrix}\begin{bmatrix}\boldsymbol{X}\end{bmatrix}\begin{bmatrix}a & b & a & c\\ a & c & -a & -b\\ a & -c & -a & b\\ a & -b & a & -c\end{bmatrix} \tag{4-27}$$

其中 $\boldsymbol{X}$、$\boldsymbol{Y}$ 分别为像素矩阵和变换系数矩阵，$\boldsymbol{A}$ 为变换矩阵。

$$a=\frac{1}{2},\quad b=\sqrt{\frac{1}{2}}\cos\left(\frac{\pi}{8}\right),\quad c=\sqrt{\frac{1}{2}}\cos\left(\frac{3\pi}{8}\right) \tag{4-28}$$

将(4-27)式转化为

$$\boldsymbol{Y}=(\boldsymbol{C}\boldsymbol{X}\boldsymbol{C}^{\mathrm{T}})\otimes\boldsymbol{E}=\left(\begin{bmatrix}1 & 1 & 1 & 1\\ 1 & d & -d & -1\\ 1 & -1 & -1 & 1\\ d & -1 & 1 & -d\end{bmatrix}\begin{bmatrix}\boldsymbol{X}\end{bmatrix}\begin{bmatrix}1 & 1 & 1 & d\\ 1 & d & -1 & -1\\ 1 & -d & -1 & 1\\ 1 & -1 & 1 & -d\end{bmatrix}\otimes\begin{bmatrix}a^2 & ab & a^2 & ab\\ ab & b^2 & ab & b^2\\ a^2 & ab & a^2 & ab\\ ab & b^2 & ab & b^2\end{bmatrix}\right) \tag{4-29}$$

其中⊗代表($\boldsymbol{C}\boldsymbol{X}\boldsymbol{C}^{\mathrm{T}}$)矩阵中的每个元素与 $\boldsymbol{E}$ 矩阵中的对应位置上的元素相乘，$d=c/b$。为了简化计算，令

$$d=\frac{c}{b}\approx 0.414\approx\frac{1}{2} \tag{4-30}$$

我们可以验证(4-29)式中的变换 $\boldsymbol{C}$ 是正交的，但要保证它的基向量是归一化的，则需要将乘数矩阵 $\boldsymbol{E}$ 中的 b 改为

$$b=\sqrt{\frac{2}{5}} \tag{4-31}$$

再将(4-29)式中 $\boldsymbol{C}$ 的第 2 和第 4 行和 $\boldsymbol{C}^{\mathrm{T}}$ 的第 2 和 4 列分别乘以 2，以避免变换中的除 2 运算产生非整数，与此同时将 $\boldsymbol{E}$ 中对应的因子除 2，则得到

$$\begin{aligned}\boldsymbol{Y}=(\boldsymbol{C}_{\mathrm{f}}\boldsymbol{X}\boldsymbol{C}_{\mathrm{f}}^{\mathrm{T}})\otimes\boldsymbol{E}_{\mathrm{f}}=&\left(\begin{bmatrix}1 & 1 & 1 & 1\\ 2 & 1 & -1 & -2\\ 1 & -1 & -1 & 1\\ 1 & -2 & 2 & -1\end{bmatrix}\begin{bmatrix}\boldsymbol{X}\end{bmatrix}\right.\\ &\left.\begin{bmatrix}1 & 2 & 1 & 1\\ 1 & 1 & -1 & -2\\ 1 & -1 & -1 & 2\\ 1 & -2 & 1 & -1\end{bmatrix}\right)\otimes\begin{bmatrix}a^2 & \frac{ab}{2} & a^2 & \frac{ab}{2}\\ \frac{ab}{2} & \frac{b^2}{4} & \frac{ab}{2} & \frac{b^2}{4}\\ a^2 & \frac{ab}{2} & a^2 & \frac{ab}{2}\\ \frac{ab}{2} & \frac{b^2}{4} & \frac{ab}{2} & \frac{b^2}{4}\end{bmatrix}\end{aligned} \tag{4-32}$$

由(4-32)式看到 4×4 变换 $\boldsymbol{C}_{\mathrm{f}}\boldsymbol{X}\boldsymbol{C}_{\mathrm{f}}^{\mathrm{T}}$ 只由整数加法、减法和移位(×2)操作完成。如果输入矩阵 $\boldsymbol{X}$ 的元素值在[−255,255](即 8 b)之内，变换系数值的范围最大可扩展 6^2 倍，因此整个变换操作可由 16 b 的整数运算完成(除了极偶然的非正常输入之外)。最后与 $\boldsymbol{E}_{\mathrm{f}}$ 相乘时需要对每个系数进行一次乘法，考虑到量化也涉及乘(除)法运算，我们可以将此合并到后面的量化过程中去，利用一次乘法完成。值得注意，虽然 4×4 变换 $\boldsymbol{C}_{\mathrm{f}}\boldsymbol{X}\boldsymbol{C}_{\mathrm{f}}^{\mathrm{T}}$ 可以由整数运算完成，但这是将它所需要的乘数部分移到量化中去而实现的，因此它不能称为严格意义上的整数变换。

(4-32)式的变换是一个正交变换，但由于 b 和 d 的变化(见(4-30)式和(4-31)式)，它并不等于(4-27)式表示的 DCT。它与 DCT 的性能相近。研究表明[8]，对于相关系数等于 0.9 的平稳高斯—马尔柯夫输入过程，(4-32)式的变换编码增益为 5.38 dB，而 DCT 的编码增益为 5.39 dB，

0.01dB 的增益差别不会造成可察觉的性能损失；对于预测误差而言，相关系数往往低于 0.9，则编码增益的差别更小，可以忽略不计。

4×4 的逆变换由下式给出：

$$\boldsymbol{X}=\boldsymbol{C}_i^{\mathrm{T}}(\boldsymbol{Y}\otimes\boldsymbol{E}_i)\boldsymbol{C}_i=\begin{bmatrix}1&1&1&1/2\\1&1/2&-1&-1\\1&-1/2&-1&1\\1&-1&1&-1/2\end{bmatrix}\left(\begin{bmatrix}\boldsymbol{Y}\end{bmatrix}\otimes\begin{bmatrix}a^2&ab&a^2&ab\\ab&b^2&ab&b^2\\a^2&ab&a^2&ab\\ab&b^2&ab&b^2\end{bmatrix}\right)\begin{bmatrix}1&1&1&1\\1&1/2&-1/2&-1\\1&-1&-1&1\\1/2&-1&1&-1/2\end{bmatrix} \tag{4-33}$$

由(4-33)式看出，变换系数首先乘以乘数矩阵 $\boldsymbol{E}_i$（与反量化一起进行），然后再进行反变换。为了减小动态范围的扩展，矩阵 $\boldsymbol{C}_i$ 和 $\boldsymbol{C}_i^{\mathrm{T}}$ 的元素取为 1 和 1/2，其中的除 2 运算可以由向右移位来完成。

2. 整数运算的量化

基本的量化操作如下式所示：

$$|Z_{ij}|=\mathrm{floor}\left(\frac{|Y_{ij}|}{Q}+f\right),f<1 \tag{4-34}$$

式中，Y_{ij} 和 Z_{ij} 分别为量化前、后的变换系数，Q 为量化步长，f 反映量化器死区宽度，floor(·)为取整。顺便指出，在实际应用中用“取整”代替“四舍五入”，往往能获得更好的主观图像质量。

在视频压缩的国际标准中，通常给出一定数量的可选的量化步长。例如表 4-2 为 H.264 规定的 52 种量化步长，表中 QP 为索引值，也称为**量化参数**，Q 为量化步长的实际大小。

表 4-2　H.264 编码器的量化步长

QP	0	1	2	3	4	5	6	7	8	9	10	11
Q	0.625	0.6875	0.8125	0.875	1	1.125	1.25	1.375	1.625	1.75	2	2.25
QP	…	18	…	24	…	30	…	36	…	42	…48	…51
Q		5		10		20		40		80	160	224

将(4-32)式中的 $\boldsymbol{E}_f$ 结合进量化，(4-34)式变为

$$|E_{ij}|=\mathrm{floor}\left(\frac{|W_{ij}|\cdot PF}{Q}+f\right) \tag{4-35}$$

式中，W_{ij} 为经($\boldsymbol{C}_f\boldsymbol{X}\boldsymbol{C}_f^{\mathrm{T}}$)变换后的系数，$PF$ 为该系数对应位置上的乘数(a^2，$ab/2$ 或者 $b^2/4$)。为了回避除法运算产生浮点数，可以将 PF/Q 乘以一个很大的数，然后再进行右移位操作，即令

$$\frac{MF}{2^q}=\frac{PF}{Q} \tag{4-36}$$

其中：

$$q=15+\mathrm{floor}(QP/6)\qquad(\text{比特}) \tag{4-37}$$

根据(4-35)式和(4-36)式，量化过程可以由下式所示的整数运算完成：

$$\begin{aligned}|Z_{ij}|&=(|W_{ij}|\cdot MF+f\cdot 2^q)\gg q\\ \mathrm{Sign}(Z_{ij})&=\mathrm{Sign}(W_{ij})\end{aligned} \tag{4-38}$$

其中≫代表二进制的右移位操作。如果将不同 QP 下对应于不同系数位置的 MF 值预先制成表，那么在对每一个变换系数进行量化时，查到该系数对应的 MF 值，只需一次乘法就可以同时完成量化和(4-32)式所要求的相乘两个运算，明显降低了计算的复杂度。式中 f 为常数，在 H.264 的参考模型软件中，对帧内和帧间编码模块 f 分别取 1/3 和 1/6。

值得注意，表 4-2 表示出 QP 每增加 6，量化步长增大 1 倍，这意味着 MF 减小 1 倍，因此在(4-

37)式中 QP 每增加 6，q 增加 1b。从(4-36)式看出，这使得我们只需要针对 QP 的前 6 个值制作 MF 表。对于 $QP>5$ 的情况，可以周期性地重复使用该表。

基本的反量化过程由下式所示：

$$Y'_{ij} = Z_{ij} \cdot Q \tag{4-39}$$

将(4-33)式所要求的相乘运算合并到反量化过程中，则得到

$$W'_{ij} = Z_{ij} \cdot Q \cdot PF \cdot 64 \tag{4-40}$$

其中 PF 为(4-33)式中 $\boldsymbol{E}_i$ 对应位置上的元素(即 a^2、ab 或 b^2)，乘以 64 是为了避免舍入误差。对 W'_{ij} 进行(4-33)式的逆变换 $\boldsymbol{C}_i^{\mathrm{T}} W C_i$，再除以 64(通过移位完成)，则可获得重建的 $\boldsymbol{X}$ 值。为了简化运算，令 $V=(Q \cdot PF \cdot 64)$，并像 MF 一样将前 6 个 $QP(0 \leqslant QP \leqslant 5)$对应于不同系数位置的 V 值预先制成表，则(4-40)式变为

$$W'_{ij} = Z_{ij} \cdot V_{ij} \cdot 2^{\mathrm{floor}(QP/6)} \tag{4-41}$$

每个变换系数的反量化和(4-33)式要求的相乘也只需一次乘法完成。当 $QP>5$ 时，V 值表可以周期性地重复使用，因为由 QP 增大引起的 Q 所增大的倍数已由 $2^{\mathrm{floor}(QP/6)}$ 表示。

顺便指出，对于 16×16 帧内编码的宏块，由于宏块中 16 个 4×4 变换后产生的 DC 系数之间存在较强的相关性，因此可以对由 4×4 个 DC 系数组成的矩阵再进行**哈达玛**(Hadamard)**变换**[9]，以进一步提高压缩编码的效率。

4.1.7 码流结构

如 4.1.1 节所述，视频序列是以帧为单位进行编码的；每一帧又被划分为条、宏块和块分别进行具有运动补偿的帧间预测和 DCT 编码，然后转换成一维的符号序列，经熵编码形成一串由 0 和 1 组成的码流。要使解码器能够从码流恢复出原来的视频序列，必须给出码流的各段与帧、条、宏块与块的对应关系，这通常是将码流按层次结构来组织，并在每个层次插入相应的头信息来完成的。图 4-18 给出了一个典型的码流层次结构图。

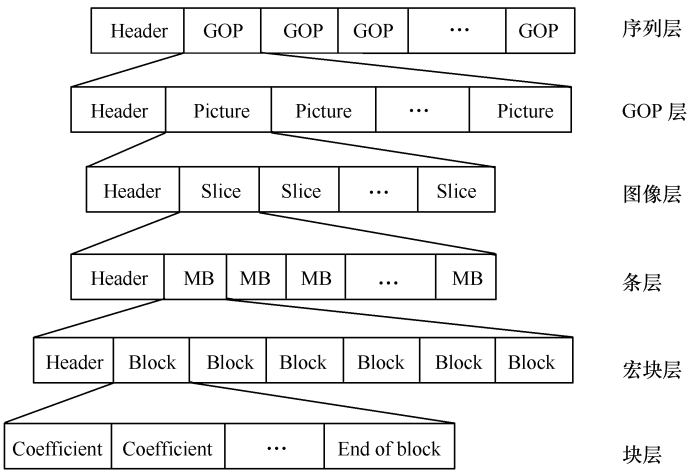


图 4-18 典型的码流层次结构

在图 4-18 中，码流共分为 6 个层次：

(1) **图像序列**。序列头给出图像分辨率、宽高比、帧率和比特率等信息；

(2) **图像组**(GOP)。它是进行随机存取的单元，其头信息是可选的，包括 GOP 的开、闭等；

(3) **图像**。它是基本的编码单元，其头信息中记录着该帧的类型(I、P 或 B)、按帧或场编码及量化表等；

(4) **条**(Slice)。一幅图像可以分成一个或多个条,在每条的开始,进行 DPCM 的参量(如 MV 和 DC 系数)的预测值都重新置到零(称为**再同步**),这可以防止解码时误差的传播。它是一帧中最大的可独立解码的单元,也称为再同步的单元。条的头信息包括条的起始位置,是否全部宏块为帧内编码等;

(5) **宏块**。它是进行运动补偿的基本单元,其头信息包括宏块的编码类型、MV 个数等;

(6) **块**。它是进行 DCT 的基本单元。

通常,国际标准并不对编码器的设计作限制,这给编码器的开发者留下了设计的余地。国际标准只规定已编码的码流必须遵循的**句法**(Syntax)、**语义**(Semantics)和一个标准解码器。所谓句法是指表述码流各层次信息的标识符的集合;语义则是句法中各符号所对应的定义。因此,只要一个厂家开发的编码器产生的码流符合国际标准,就能被其他厂家开发的标准解码器正确解码。

4.2 基于对象的视频编码

4.2.1 编码器与解码器

基于对象的编码意味着以**可视对象**(Visual Object)为基本单元进行编码和解码。可视对象可以是矩形的视频帧、视频场景中的某一物体或者是由计算机生成的 2D 或 3D 图形等。当可视对象是视频帧或帧内物体时,称为**视频对象**(Video Object,VO)。如图 4-19 所示,VO 由时间上连续的若干帧中的同一物体组成,具有一定的生命周期;而某一时刻的 VO,即某一帧中的 VO 称为**视频对象平面**(VO Plane,VOP)。一个视频场景可以由一个背景对象和若干个前景对象构成。针对每个对象分别进行编码比针对整个视频帧编码具有更大的灵活性,例如,根据重要性不同每个对象可以不同的质量或帧率进行编码;可以将不同渠道摄取的自然对象以及计算机生成的对象组合在一个场景中;允许用户通过交互手段对场景的构成进行操作,如从场景中删掉某个对象等。

图 4-20 给出了一个简化的基于对象的视频编码器框图。与图 4-1 相类似,它也采用了包括帧间预测编码、DCT、量化和熵编码结合的混合编码方案。不同的是,这里的编码针对 VOP 进行,图中有三个预测器(假设有三个对象),对每个 VOP 独立地进行编码,在输出端再将各 VO 生成的码流复用成一个流。根据编码方式可以将 VOP 分为三类:I-VOP(帧内编码),P-VOP(前向预测编码)和 B-VOP(双向预测编码)。

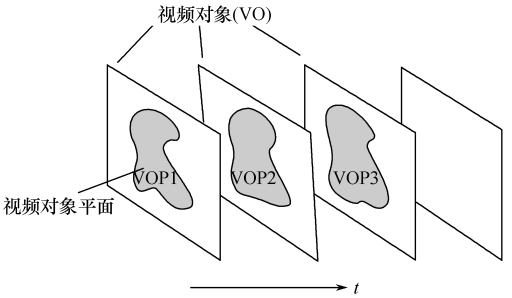


图 4-19 VO 和 VOP

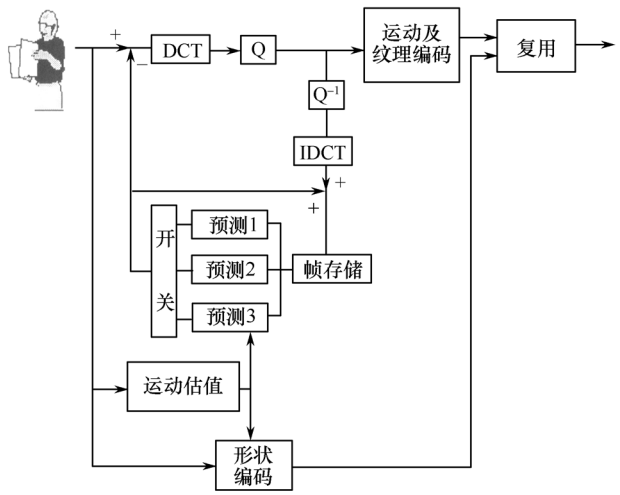


图 4-20 基于对象的视频编码器简化框图

由于 VOP 可以是任意形状的，因此除了运动信息（MV）和以 DCT 系数表示的纹理（预测误差）信息外，还需要将形状信息传递给解码端。图中所示的形状编码是图 4-1 中所没有的。同时 VOP 形状的任意性也要求我们对以前讨论的基于矩形块的运动估值和运动补偿，以及块 DCT 等算法进行相应的扩展。

解码器完成编码的逆过程，即从接收码流中分离出各 VOP 码流，然后对每个 VOP 解码，最后组合成一帧图像。

4.2.2 任意形状 VOP 的预测编码

对任意形状 VOP 进行运动估计和运动补偿时，首先根据 VOP 的形状将 VOP 用一个矩形框套住，然后对矩形框中的宏块（或 8×8 的块）分别进行处理。为了说明可能遇到的不同情况，图 4-21 给出了当前帧 VOP 和参考帧 VOP 的一个例子，其中阴影部分表示 VOP，空白部分表示 VOP 之外的区域。图中 MB_1 全部包含在当前 VOP 中，而它的匹配块却有部分落在参考 VOP 之外；宏块 MB_2 和 MB_3 为边缘块，它们都有一部分在当前 VOP 之外，它们的匹配块也有部分在参考 VOP 之外。

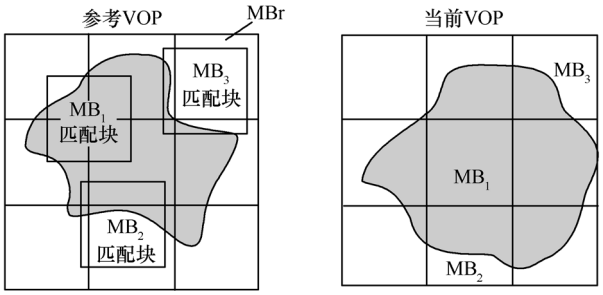


图 4-21　当前帧和参考帧的 VOP

在进行运动估计时，我们仅对当前 VOP 内部的点计算预测误差 SAD 值，即对全部在当前 VOP 之内的块（如图中 MB_1）按一般矩形块的方法计算；而对边缘块（如 MB_2 和 MB_3）则按下式计算 SAD 值：

$$\mathrm{SAD}=\sum_{i=1}^{M}\sum_{j=1}^{N}\left|b_k(i,j)-b_{k-1}(i+mv_x,j+mv_y)\right|\cdot(!(\mathrm{Alpha}_k(i,j)==0)) \tag{4-42}$$

其中 k 和 $k-1$ 分别代表当前帧和参考帧，mv_x 和 mv_y 分别代表运动矢量的水平和垂直分量，$\mathrm{Alpha}_k(\cdot)$ 代表当前 VOP 的形状信息，它是一个非 0 即 1 的二值量，等于 1 时代表 VOP 内部，等于 0 时为 VOP 外部。

如果当前块的匹配块落在只包含部分参考 VOP 的宏块内，或者落在完全不包含参考 VOP 的宏块内（如图 4-21 中 MB_3 的匹配块有一部分落在完全不包含参考 VOP 的宏块 MBr 中），那么我们还需要对落在参考 VOP 之外的区域进行填充，以获得较小的预测误差。我们首先对包含部分参考 VOP 的宏块进行水平填充，如果欲填充行的右边和左边都有已赋值的点，即取这两点的平均值，否则哪边有已赋值的点就取最邻近点的值，进行填充。如果两边都没有已赋值的点，则等待垂直填充。图 4-22(b)表示对(a)所示的块进行水平填充后的结果，图中粗线表示参考 VOP 的边界，深色块为参考 VOP 的内部。在水平填充之后进行垂直填充，如果欲填充列的上边和下边都有赋值的点，则取这两点值的均值，否则哪边有赋值点就取最邻近点的值，进行填充。图 4-22(c)给出对(b)进行垂直填充后的结果。对于在匹配块之内又完全不在参考 VOP 内的宏块（如图中 MBr）也需要进行填充。如果只有一个水平或垂直的参考 VOP 边缘块与它相邻，采用最邻近点的值对它

进行填充；如果该宏块有 1 个以上的相邻边缘块，按左、上、右和下的优先顺序选择一个最邻近点的值进行填充；如果没有边缘块与之邻接，则用 2^{N-1} 填充，N 是每像素的比特数，当 $N=8$ 时，填充值为 128。

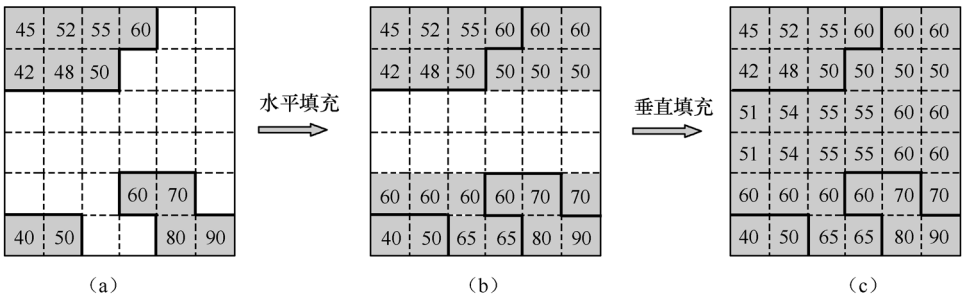


图 4-22　参考区域的填充

4.2.3　任意形状 VOP 的纹理编码

1. 块填充方法

对于完全在当前 VOP 之内的宏块，其纹理编码与基于帧的传统编码方式相同，即采用 8×8DCT、量化和熵编码。对于当前 VOP 的边缘宏块，由于落在 VOP 之外的像素值没有定义，因此需要先进行填充再进行 8×8DCT。在解码时，解码器会根据形状信息删除 VOP 之外（即填充）的像素，因此填充的具体数值并不影响 VOP 的解码，而主要考虑的是，使空间域内像素值的变化比较平缓，DCT 后能量尽可能地集中在低频部分，以提高编码效率。例如，对帧内编码的边缘块，可以填充 2^{N-1}，或者采用图 4-22 所示的重复填充技术填充。对于帧间编码的残差块，一般用 0 来填充即可。

低通外插（Low-Pass Extrapolation，LPE）法是另一种进行块填充的技术，它可以使 8×8 块更为平滑，DCT 变换后能量更集中，从而提高编码效率。LPE 填充算法如下。

（1）计算属于 VOP 区域的像素 f 的算术平均值：

$$m=\frac{1}{N}\sum_{(i,j)\in VOP}f(i,j) \tag{4-43}$$

式中，N 为 VOP 内的像素数。

（2）将 m 赋值给块内不属于 VOP 区域的像素。

（3）再对块内 VOP 区域外的像素点用其邻近的 4 个像素点进行平均滤波操作，从块的左上角开始沿行一直到块的右下角结束：

$$f(i,j)=[f(i,j-1)+f(i-1,j)+f(i+1,j)+f(i,j+1)]/4 \tag{4-44}$$

上式中，如果某像素点在块的外部，则这些像素点不使用，相应的除以 4 变成实际使用的像素点数。第三步可以根据需要进行多次迭代。

无论采用什么样的填充方法，填充后宏块中的 16×16 个像素都需要参与编码，这对于边缘块来讲降低了编码效率。下面介绍的方法则对此进行了改进，它只对 VOP 之内的像素进行纹理编码。

2. 形状自适应 DCT

形状自适应 DCT（Shape-adaptive DCT），简称 **SA-DCT**，利用一组预先定义好的一维 DCT 来有效地进行任意形状 VOP 的变换编码。我们以图 4-23(a)所示的 8×8 块为例来说明它的工作过程，图中深色块代表在 VOP 内的像素，白色块为 VOP 之外的像素。前向 SA-DCT 的步骤如下：

（1）将 VOP 之内的像素 X 纵向移动到块的顶端[见图 4-23(b)]；

(2) 对每一列分别进行 n 点 1D DCT，n 值由该列中 VOP 内像素的个数决定[见图 4-23(c)]；

(3) 将上述变换后的值移到块的左端[见图 4-23(d)]；

(4) 对每一行进行 m 点 1D DCT，m 值由该行中 VOP 内像素的个数决定[见图 4-23(e)]。

在接收端进行上述步骤的逆操作，即可重建 8×8 的原始块。SA-DCT 比一般 DCT 复杂，但它可以提高边缘块的编码效率。

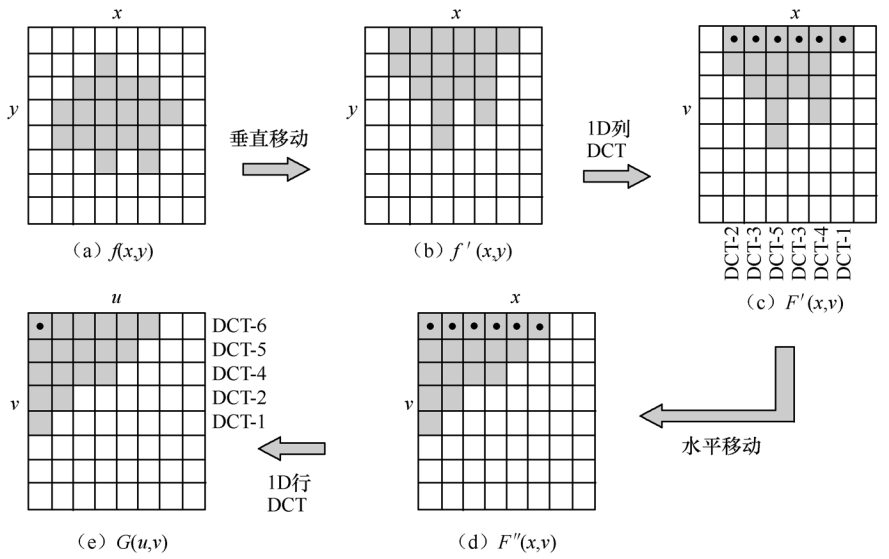


图 4-23　SA-DCT

4.2.4　形状编码

VO 形状信息的编码有两类，一类是二值形状信息，另一类是灰度形状信息。二值形状信息是用 0、1(实际是 0 和 255)的方式表示 VOP 的形状，0 表示非 VOP 区域，常称为**透明**区域，而 1 则表示 VOP 区域，称为**非透明**区域。灰度形状信息可取值 0～255，类似于图形学中的 Alpha 平面的概念，0 表示非 VOP 区域(即透明区域)，1～255 表示 VOP 区域的透明程度，255 表示完全不透明的 VOP 区域。灰度形状信息可以使前景 VO 更平滑地叠加到背景上去。

形状信息通常用位图法表示。将 VOP 用一矩形“边框”套住，边框满足一定的条件：(1)边框的长、宽都是 16 的倍数；(2)边框左上角点的绝对位置坐标必须是偶数；(3)必须使边框尽可能地小。这样对 VOP 的形状信息编码变为对该边框形成的矩形位图进行无损或有损编码。用边框矩阵位图法表示形状信息，可以在兼顾基于 16×16 宏块的运动估计的同时，降低形状信息的数据量，既提高了压缩效率，又降低了计算量。

1. 二值形状编码

二值形状编码是基于 16×16 的 **BAB**(Binary Alpha Block)块的，VOP 的边框矩阵位图由整数个 BAB 组成。根据 VOP 不同的类型，其编码方式也有所不同。

I-VOP 的 BAB 块有三种类型：BAB 块在 VOP 区域内(全 1)、BAB 块在 VOP 区域外(全 0)和 BAB 块跨越 VOP 边界。对跨边界的边缘 BAB 块进行基于上下文的算术编码[9]。

P-VOP 和 B-VOP 在 BAB 编码前需进行运动补偿。BAB 的运动矢量可以和 VOP 本身纹理块的运动矢量不相同。BAB 运动补偿的方法与纹理块的运动补偿方法相同，但只作整像素精度和整个 16×16 块(1 个运动矢量)的补偿。帧间编码 BAB 也采用基于上下文的算术编码，不过它与帧内编码 BAB 计算上下文的方法不同。

此外，在码率控制过程中可能会改变 VOP 二值形状图的分辨率。分辨率转换一般采用1∶2下采样。下采样后的二值图通过内插可得到原分辨率大小的形状图，为了避免过大的失真，只有当重构的形状图与原形状图的误差小于一定的阈值，才可以进行分辨率转换。

2. 灰度形状编码

VOP 的灰度形状图分为两部分：轮廓与 Alpha 值。其中轮廓图是二值的，0 表示透明区，1 表示半透明和不透明的 VOP 区域；Alpha 则是灰度图，表示 VOP 区域的透明程度。轮廓图采用与 BAB 相同的二值形状编码，Alpha 图则采用 4.2.3 节介绍的方法进行纹理编码。

4.2.5 码流结构

图 4-24 给出一个基于对象的视频码流的层次结构。最上层是可视对象序列 **VS**(Visual object Sequence)，一个 VS 可以包括一个或几个可视对象，这些对象可以是视频对象，也可以是计算机产生的 2D/3D 网格对象等。在这里我们只考虑视频对象，并简称其为 **VO**。

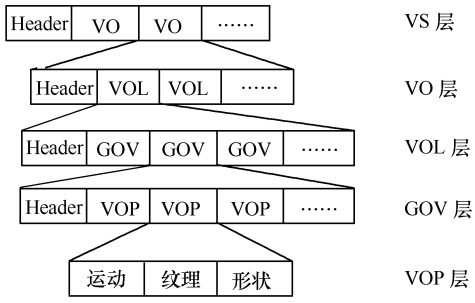


图 4-24 基于对象的码流层次结构

一个 VO 可以有一个或若干个层(Video Object Layer，**VOL**)，例如基本层、增强层等。在可伸缩性编码中(见 10.2 节)，每一个层是 VO 在某一时域分辨率、空域分辨率，或某个信噪比下的表示；在普通的非可伸缩性编码中，VO 只有一个层。**GOV**(Group of VOPs)为一组 VOP，类似于基于帧编码中的 GOP，为随机接入码流提供入口。图示结构的最低层次为 VOP，它是 VO 在某一时刻的表示，即某一帧的 VO。VOP 包含它编码后的运动、纹理和形状数据。每一个编码层次加上说明该层次参数的头，编码后的数据和头信息逐层次复用构成可视对象序列的码流。

4.3 恒定速率编码器的速率控制

如 4.1.1 节所述，编码器所产生的码流其数据率一般是变化的，为了适应恒定速率(CBR)信道的要求，在编码器和信道之间需要设置一个缓存器。当码流的速率高于信道的传输速率时，缓存器会越来越满；而码率低于信道速率时，缓存器则越来越空。为了防止缓存器溢出或变空，需要对压缩码流的速率进行控制。这通常是通过图 4-1 中的反馈支路，改变量化器的量化步长来实现的。量化步长加大，码率下降，但图像质量也会因此而下降；而量化步长减小，则导致码率上升，图像质量随之变好。速率控制的基本问题就是在尽可能地保证图像质量稳定的条件下，使压缩码流的速率适应恒定速率信道的要求。

4.3.1 高码率应用的速率控制(TM 5)

视频编码标准 MPEG－2 的测试模型(Test Model 5，TM5)给出了一个高码率下的典型速率控制算法，它分为目标比特分配、码率调整和自适应量化三个步骤。

1. 目标比特分配

所谓**目标比特分配**就是在每帧(或每个宏块)编码之前，根据一定的原则为其规定好编码的比特数，称为目标比特数。分配的基本原则是，编码产生的码流平均速率与指定的信道速率相匹配。在 TM5 中，除上述原则外，还考虑到帧的编码类型(I、P、B)和图像内容的复杂程度。由于在一组

图像(GOP)中，通常 I 帧产生的比特数多，而 B 帧编码后的比特数较少，P 帧居中，为了使图像质量在帧间不出现显著的变化，分配给 I、P、B 图像的比特数(目标比特数) T_i、T_p 和 T_b 满足 $T_i > T_p > T_b$。此外，图像复杂程度由一个全局复杂度变量 $X_x = S_x \cdot Q_x$，$(x = i, p, b)$ 来衡量，其中 S_x 为前一个 x 帧(即 I 帧、P 帧或 B 帧)实际产生的比特数，Q_x 为该帧编码时所有宏块实际使用的量化参数的平均值。由于图像内容在时间上的相关性，我们可以用 X_x 来估计当前帧的复杂程度。

假设在某个 GOP 开始时，可分配的比特数为 $R = N \cdot R_c / f_f + R'$，其中 N 为 GOP 中图像的总数，f_f 为帧率，R' 为上一个 GOP 编码所剩余的比特数。当每编完本 GOP 中的一帧图像时，可分配的比特数更新为 $R = R - S_x$，然后按下式在余下的帧中进行分配：

$$T_i = \max\left\{\frac{R}{1 + N_p \dfrac{X_p}{X_i K_p} + N_b \dfrac{X_b}{X_i K_b}},\quad R_c/8f_f\right\} \tag{4-45}$$

$$T_p = \max\left\{\frac{R}{N_p + N_b \dfrac{X_b \cdot K_p}{X_p \cdot K_b}},\quad R_c/8f_f\right\} \tag{4-46}$$

$$T_b = \max\left\{\frac{R}{N_b + N_p \dfrac{X_p \cdot K_b}{X_b \cdot K_p}},\quad R_c/8f_f\right\} \tag{4-47}$$

式中，N_P、N_b 分别为余下的 P 帧和 B 帧的个数，K_p 和 K_b 为由量化矩阵所确定的控制松紧的系数。以(4-46)式为例，$N_b \cdot X_b / X_p$ 可以看成是将剩余的 B 帧折算成 P 帧时的等效帧数。$\max\{\cdot\}$ 符号中后一项给出每帧目标比特数的下限。

2. 码率调整

所谓**码率调整**是指在每一帧的编码过程中，通过动态调整每个宏块的量化步长，使该帧实际产生的比特数接近预分配的目标比特数。为了达到这个目的，我们假设了三个虚拟缓存器，每一个用于一种类型(I、P 或 B)图像的编码输出，编码器输出缓存器的实际充满程度是三个虚拟缓存器充满程度之和。以其中一个虚拟缓存器为例，它的容量等于该帧所分配的目标比特数 T_x。图 4-25 表示出缓存器充满程度随已编宏块数的变化曲线。图中虚线表示 T_x 按宏块均匀分配时缓存器占有量增长的情况，实线表示实际占有量的变化。在进行第 j 个宏块的编码之前，检查缓存器占有量 d_j^x 超过预分配值的大小：

$$d_j^x = d_0^x + B_{j-1} - \frac{T_x}{\text{MB_ count}}(j-1) \tag{4-48}$$

式中，d_0^x 为缓存器的初始占有量，即前面各帧编码所产生的数据超过预分配值的大小，B_{j-1} 为本帧前$(j-1)$个宏块编码所实际产生的比特数，MB_count 为本帧的总宏块数。d_j^x 越大，表示后面的宏块应该使用比平均分配值更小的比特数，才能使本帧图像编码产生的总比特数接近目标值 T_x。因此，第 j 个宏块的参考量化参数取为：

$$Q_j = \frac{d_j^x \cdot 31}{r} \tag{4-49}$$

式中，$r = 2R_c / f_f$，为响应参数。

3. 自适应量化

考虑到视觉的掩蔽效应，某个宏块实际使用的量化步长是由参考量化参数 Q_j 和该宏块图像的活跃程度(即亮度变化的剧烈程度)两个因素最后确定的。将 16×16 的亮度块划分成 4 个 8×8 的子块，定义该亮度块的**空间活动度**(Activity Function)为

$$act_j = 1 + \min(\text{Var_sblk}) \tag{4-50}$$

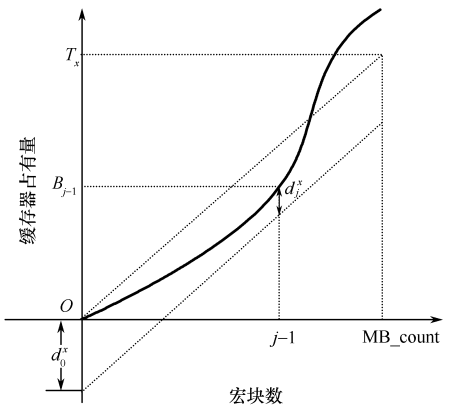


图 4-25　虚拟缓存器占有量随已编宏块数目的变化曲线

式中，Var_sblk 为该宏块中的子块像素值的方差。将 act_j 归一化，得到

$$N_act_j = \frac{2 \cdot act_j + avg_act}{act_j + 2 \cdot avg_act} \tag{4-51}$$

式中，avg_act 为前一帧的 act_j 的均值。用 N_act_j 对从比特率要求导出的参考量化参数 Q_j 加权，得到该宏块的量化步长为

$$(quantization_scale)_j = Q_j \cdot N_act_j \tag{4-52}$$

当 $act_j \gg avg_act$ 时，表示图像亮度变化剧烈，$N_act_j \approx 2$，根据视觉的掩蔽效应，在这种情况下可以采用较大的量化步长；反之，$act_j \ll avg_act$，$N_act_j \approx \frac{1}{2}$，表示图像亮度变化平坦，量化噪声的能见度较高，则采用较小的量化步长。

TM5 能够满足使编码器输出码流的速率与信道速率相匹配的要求，而且由于它的比特分配策略比较合理，码率调整较为精细，并考虑了人的视觉特性，因此在限定的速率下重建图像具有较高的主观质量。但是，TM5 的速率控制策略仍有不足之处，例如一帧内所有宏块的预分配比特数是相同的，并未考虑宏块的类型；目标比特的分配和空间活动度的归一化均基于上一帧的信息，不能很好地处理场景的变换等。

4.3.2　速率控制中的率－失真模型

如何根据预分配的比特数选择合适的量化步长是速率控制中的一个基本问题，而量化步长既影响解码图像失真，也影响输出码率。我们根据 3.3 节已知，速率和失真之间的关系由率失真函数决定，因此根据率失真函数建立模型，并将其引进速率控制是一个合理的思路。

研究者已经建立起多种用于速率控制的率－失真模型，下面我们介绍应用较为广泛的两种。

1. 速率模型和失真模型

假设经运动补偿后残差的 DCT 系数是近似互不相关的，且符合方差为 σ^2 的拉普拉斯分布。由于 DCT 是一个正交变换，因此残差帧像素值的方差也为 σ^2。我们再假设 DCT 系数经均匀量化后熵编码所产生的数据率（每个符号的比特数）$R(Q)$ 近似等于 DCT 系数的熵 $H(Q)$（见 3.9.3 节），其中 Q 为系数的量化步长。

根据文献[10]，$H(Q)$ 近似为

$$H(Q) = \begin{cases} \frac{1}{2}\log_2(2e^2\frac{\sigma^2}{Q^2}) & \frac{\sigma^2}{Q^2} > \frac{1}{2e} \\ \frac{e}{\ln 2}\cdot\frac{\sigma^2}{Q^2} & \frac{\sigma^2}{Q^2} \leqslant \frac{1}{2e} \end{cases} \tag{4-53}$$

式中，e 为常数。在只考虑低码率和高失真（PSNR 值$<$40dB）的情况下，Q 值较大，满足 $\sigma^2/Q^2 < 1/2e$，因而我们有

$$R(Q) \approx H(Q) \approx K\frac{\sigma^2}{Q^2} \quad (比特/符号) \tag{4-54}$$

式中，K 为常数。在 DCT 系数满足拉普拉斯分布时，$K = e/\ln 2$，但考虑到上式是一个近似关系，K 通常通过实验数据来估计，且 $0 \leqslant K \leqslant e/\ln 2$。根据(4-54)式得到的**速率模型**如下：

$$B_i = A(K\frac{\sigma_i^2}{Q_i^2} + C) \quad (比特) \tag{4-55}$$

式中，B_i 为编码第 i 个单元（如第 i 个宏块）的比特数；A 为该单元的样值数，对于一个宏块，$A=16$

×16；AC 为编码该单元的开销，例如运动矢量和头信息等占用的比特数。当将亮度和色差信号都考虑在内时，σ_i^2 按下式计算[10]

$$\sigma_i^2 = \frac{1}{A}\sum_{j=1}^{A+A'}[P_i(j)-\bar{P}_i]^2 \tag{4-56}$$

式中，A'为色差信号样值的个数，$P_i(j)$为第 i 个单元中第 j 个样值，$\bar{P}_i$ 为样值的均值。在这里我们假设了亮度和色差信号的均值相等。因为对于残差块而言，二者的均值都很小，因而该假设成立。

对于量化步长为 Q 的均匀量化器，当假设在每个量化区间内信源近似为均匀概率分布时，即

$$p(x)=\begin{cases}\dfrac{1}{Q} & |x|<\dfrac{Q}{2}\\ 0 & \text{其他}\end{cases} \tag{4-57}$$

根据(3-132)式得到均方量化误差

$$\sigma_e^2 = \frac{Q^2}{12} \tag{4-58}$$

如果考虑每个单元对失真的贡献不相同，则得到**失真模型**如下：

$$D = \frac{1}{N}\sum_{i=1}^{N}\alpha_i^2\frac{Q_i^2}{12} \tag{4-59}$$

式中，N 为一帧中宏块(单元)的个数，Q_i 为第 i 个宏块的量化步长，α_i 为第 i 个宏块的权重。

2. 基于速率模型和失真模型的优化

在速率控制中，希望在给定的目标比特数下找到最佳的量化步长 Q^*，以使产生的失真最小。因此利用式(4-55)和(4-59)式给出的速率模型和失真模型，我们建立如下的优化问题：

$$\begin{aligned}Q_1^*\cdots Q_N^* &= \operatorname{Arg}\min_{Q_1\cdots Q_N}\frac{1}{N}\sum_{i=1}^{N}\alpha_i^2\frac{Q_i^2}{12}\\ \text{Subject to}\quad B &= \sum_{i=1}^{N}B_i\end{aligned} \tag{4-60}$$

式中，B 为给定的当前帧编码的总比特数。将(4-55)式代入上式，并利用求条件极值的拉格朗日乘数法，得到

$$Q_1^*\cdots Q_N^*,\lambda^* = \operatorname{Arg}\min_{Q_1\cdots Q_N,\lambda}\left\{\frac{1}{N}\sum_{i=1}^{N}\alpha_i^2\frac{Q_i^2}{12}+\lambda\left[\sum_{i=1}^{N}A\left(K\frac{\sigma_i^2}{Q_i^2}+C\right)-B\right]\right\} \tag{4-61}$$

对上式求偏导并令其等于 0，得到最佳量化步长 Q_i^* 为

$$Q_i^* = \sqrt{\frac{AK}{(B\text{-ANC})}\cdot\frac{\sigma_i}{\alpha_i}\sum_{k=1}^{N}\alpha_k\sigma_k}\qquad (i=1,2,\cdots,N) \tag{4-62}$$

在低码率应用中，权重 α_i 可以用下式计算[10]：

$$\alpha_i=\begin{cases}2\dfrac{B}{AN}(1-\sigma_i)+\sigma_i & \dfrac{B}{AN}<0.5\\ 1 & \text{其他}\end{cases} \tag{4-63}$$

将(4-62)式代入(4-55)式，可以看到在使用最佳量化步长 Q^* 时，分配给每个宏块的比特数 B_i 为

$$B_i = \frac{(B\text{-ANC})\alpha_i\sigma_i}{\sum_{k=1}^{N}\alpha_k\sigma_k}+AC \tag{4-64}$$

3. 二次率－失真模型

这里介绍另一种用于速率控制的率-失真模型。我们仍假设经运动补偿后残差的 DCT 系数是近似互不相关的，且符合拉普拉斯分布：

$$p(y)=\frac{\alpha}{2}e^{-\alpha|y|}\qquad -\infty<y<\infty \tag{4-65}$$

并假设失真度量为

$$d(y,\tilde{y})=|y-\tilde{y}| \tag{4-66}$$

其中 $\tilde{y}$ 为 y 的重建值，则率－失真函数可以用下面的解析函数表示[11]

$$R(D)=\ln(1/\alpha D) \tag{4-67}$$

其中

$$D_{\min}=0,\quad D_{\max}=1/\alpha,\quad 0<D<\frac{1}{\alpha}, \tag{4-68}$$

将(4-67)式进行泰勒级数展开，得到

$$\begin{aligned}R(D)&=(\frac{1}{\alpha D}-1)-\frac{1}{2}(\frac{1}{\alpha D}-1)^2+R_3(D)\\&=-\frac{3}{2}+\frac{2}{\alpha}D^{-1}-\frac{1}{2\alpha^2}D^{-2}+R_3(D)\end{aligned} \tag{4-69}$$

式中，$R_3(D)$为余项。根据上式提出的率－失真模型如下[11]：

$$R=X_1Q^{-1}+X_2Q^{-2} \tag{4-70}$$

式中，Q 为量化步长，X_1 和 X_2 为模型参数。(4-70)式称为**二次**(Quadratic)**率-失真模型**。根据这个模型，已知 R 可以直接计算出 Q。

在推导上述模型时隐含地假设了每一帧图像的复杂度是近似相同的，因此同样的比特数下，每帧图像的失真相同。如果将每帧图像的复杂度实际并不相同这一因素考虑在内，(4-70)式变成

$$\frac{R}{S}=X_1Q^{-1}+X_2Q^{-2} \tag{4-71}$$

式中，S 为表示图像复杂度的一个参量，例如经运动补偿后残差的 MSE，或 MAD。

(4-70)式的另一个不足是，没有将编码辅助信息(如 MV、编码模式及形状信息等)所需的比特数考虑在内。为此而改进的模型如下式所示：

$$\frac{R-H}{S}=X_1Q^{-1}+X_2Q^{-2} \tag{4-72}$$

式中，H 为对辅助信息进行编码所需的比特数。

在实际应用中使用上述率－失真模型，我们需要知道模型参数 X_1 和 X_2。这可以利用若干已知数据点对(R_i,Q_i)，$i=1,2,\cdots,n$，对 X_1 和 X_2 进行估计来得到，这里 Q_i 和 R_i 分别为已编码帧(或宏块)实际使用的量化步长和实际产生的比特数[对于(4-72)式，速率值使用$(R_i-H_i)/S_i$]。利用**最小二乘法**，在最小均方误差意义下，使(4-70)式与观测值(R_i,Q_i)有最佳拟合的模型参数如下：

$$\begin{bmatrix}X_1\\X_2\end{bmatrix}=(\boldsymbol{Z}^{\mathrm{T}}\cdot\boldsymbol{Z})^{-1}\cdot\boldsymbol{Z}^{\mathrm{T}}\cdot\boldsymbol{Y} \tag{4-73}$$

式中，$\boldsymbol{Z}_{n\times 2}=[1,1/Q_i]$，$\boldsymbol{Y}_{n\times 1}=[Q_i\cdot R_i]$。当获得模型参数 X_1 和 X_2 之后，根据(4-70)式或(4-72)式所示的模型和当前帧(或当前宏块)分配的目标比特数，就可以计算出当前帧(或宏块)应该使用的量化步长 Q。

4.3.3 低延时应用的速率控制(TMN8)

低延时应用的速率控制算法很多，例如 H.263 和 MPEG－4 的有关算法，本节仅以 H.263 测试模型所建议的 TMN8(Test Model，Near term，Version 8)为例介绍这类算法的特点。在低延时应用中，视频序列一般采用 $IPP\cdots P$ 的结构。由于要求有较低的延时，编码器的缓存器容量不能过大，因而在低延时应用中速率控制更为困难，防止缓存器上溢和下溢是需要特别重视的问题。在必要的时候，可以采取**跳帧**(不编码该帧)的措施避免缓存器上溢。此外，为了更精确地进行码率调整，这类算法多通过率－失真模型来计算量化参数。

TMN8 包括帧级速率控制和宏块级速率控制两部分。帧级速率控制属于一种较粗的速率控制，其主要作用是根据缓冲区状态及信道速率为每帧预分配合适的比特数；而宏块级速率控制则更精细一些，其主要作用是在使各宏块编码总比特数大致等于该帧预分配比特数的前提下，失真尽可能地小。

1. 帧级速率控制

TMN8 的帧级速率控制主要分为以下两步。

(1) 缓冲器状态及跳帧控制

进行当前帧编码之前，必须先计算缓存器的占有量 W：

$$W = \max(W_{\text{prev}} + B' - R/F, \quad 0) \tag{4-74}$$

式中，W_{prev} 为前一帧编码前缓存器的占有量，B' 为前一帧编码实际所产生的比特数，R 为信道速率，F 为帧率。缓存器预设门限值 $M=R/F$，一旦缓存器占有量超过门限值，则进行跳帧以避免缓冲区上溢。由此可见，缓存器引入的最大延时为 M/R。

(2) 比特预分配

根据上步计算所得到的缓存器状态，进行当前帧的比特分配：

$$B = \frac{R}{F} - \Delta \tag{4-75}$$

式中

$$\Delta = \begin{cases} W/F & W > bM \\ W - bM & \text{其他} \end{cases} \tag{4-76}$$

b 为常数，其默认值为 0.1。Δ 是一个不大的数，当缓存器占有量大于门限值的 10%时，少分配一些比特；否则多分配一些比特。bM 决定了目标缓存器延时。

2. 宏块级速率控制

在宏块级速率控制中，TMN8 采用了 4.3.2 节中介绍的速率模型和失真模型，具体算法步骤大致分如下三步。

(1) 计算当前宏块的有关参数

根据(4-56)式计算第 i 个宏块的标准差 σ_i；如果 P 帧中有帧内编码宏块，该块用像素值计算标准差，并将结果除以 3。除 3 是一个经验值，一般具有同样方差的帧内编码块比帧间编码块产生的比特数少一些，因为帧内编码块像素间的相关性要大一些。然后，根据(4-63)式计算第 i 个宏块的编码权重 α_i。

计算除开销比特外编码当前宏块剩余可用的比特数 L_i：

$$L_i = \widetilde{B}_i - AN_iC \tag{4-77}$$

式中，$A=16\times16=256$；$\widetilde{B}_i$ 为编码当前宏块时该帧所剩比特数；$N_i=N-i+1$ 为编码当前宏块时剩余宏块数。

计算剩余宏块的加权能量和 S_i：

$$S_i = \sum_{k=i}^{N} \alpha_k \sigma_k \tag{4-78}$$

(2) 计算量化参数

当 $L_i>0$ 时，根据(4-62)式，可得当前宏块的最佳量化步长：

$$Q_i^* = \sqrt{\frac{256K\sigma_i S_i}{L_i \alpha_i}} \tag{4-79}$$

在 H.263 中，量化步长 Q 与量化参数 QP(索引值)有如下关系：

$$QP = \text{round}\{Q^*/2\} \tag{4-80}$$

QP 取值范围为{1,…,31},即有 31 种量化步长。为了使视觉质量较为平滑,一般要求相邻宏块间 QP 之差的绝对值不超过 2,所以做如下处理:

$$QP = \max\{QP_{\text{prev}} - 2, \min[QP_{\text{prev}} + 2, \text{round}(Q_i^* / 2)]\} \tag{4-81}$$

式中,QP_{prev}表示上一个宏块的量化参数。

对于 $L_i \leqslant 0$,则取 $QP = QP_{\text{prev}} + 2$。获得 QP 后,即可以对当前宏块进行编码。值得指出,在极低比特率时,一帧内使用同一个 QP 可以无需传输表示宏块量化步长变更的参数 $DQUANT$,因而可能有更高的编码效率。

(3) 模型参数 K 和 C 的更新

参数 K 和 C 的初始化值如下:

$$K = K_1 = 0.5, C = C_1 = 0, \text{若当前帧为初始图像}$$

$$K = K_1 = K_{prev}, C = C_1 = C_{prev}, \text{其他情况}$$

编码完当前宏块后,更新参数 K 和 C,计算步骤如下。

① 首先根据(4-55)式计算当前宏块实际的参数值:

$$\hat{K} = \frac{T'_{LC,i}(2QP)^2}{256\sigma_i^2}, \qquad \hat{C} = \frac{T'_i - T'_{LC,i}}{256} \tag{4-82}$$

式中 T'_i 为第 i 个宏块实际编码比特数,$T'_{LC,i}$ 为其中编码亮度和色度信息所用比特数。

② 然后用递归方法计算当前帧中已编码宏块的参数均值:

$$\widetilde{C}_i = \frac{(i-1)\widetilde{C}_{i-1} + \hat{C}}{i} \tag{4-83}$$

若 $0 < \hat{K} \leqslant e/\ln 2$

$$j = j + 1, \quad \widetilde{K}_j = \frac{(j-1)\widetilde{K}_{j-1} + \hat{K}}{j} \tag{4-84}$$

式中,j 为 DCT 系数符合 Laplace 分布的宏块个数,即 $\hat{K}$ 在$(0, e/\ln 2]$之内的宏块个数。

③ 更新下一宏块编码所用的模型参数:

$$K = \frac{i\widetilde{K}_j + (N-i)K_1}{N} \tag{4-85}$$

$$C = \frac{i\widetilde{C}_i + (N-i)C_1}{N} \tag{4-86}$$

在上式中,当编码前几个宏块时,由于 i 小,当前帧 K 和 C 的均值不具有代表性,因此更新值倾向于前一帧的参数值(K_1、C_1);当 $i \to N$ 时,更新值趋向于当前帧的参数均值。

4.3.4 基于率—失真优化的速率控制(JVT—G012)

如 4.1.5 节所介绍,在 H.264 中采用率失真优化(RDO)进行编码模式选择,这使得它的速率控制变得很复杂。因为在基于模型的速率控制中需要了解有关残差的信息,如 MAD 或方差(见 4.3.2 节),以决定合适的量化参数,而残差信息只有在进行 RDO 之后才能获得;在另一方面,进行 RDO 又必须先已知量化参数以决定拉格朗日参数 λ,以及对失真和码率进行估计(见 4.1.5 节),这就构成了一个鸡生蛋、蛋生鸡的悖论。图 4-26 给出将速率控制和 RDO 嵌套在一起的示意图,其计算之复杂使之不能在实际中采用。

在本节中我们将通过 JVT-G012 所建议的速率控制算法给出解决上述问题的一个例子。在这个算法中,定义了一个称为**基本单元**的编码单元,它可以是一帧、一场、一个条或一个宏块。在基本单元内使用相同的量化参数 QP,其值由 4.3.2 节介绍的二次率－失真模型确定。由于在进行 RDO 之前还没有得到残差信号,该算法提出一个**线性模型**从前一帧同位基本单元的 MAD 值对当

前帧基本单元的 MAD 值进行估计。QP 确定之后，对当前基本单元中的每个宏块进行 RDO 的模式选择。

假设视频序列的 GOP 中只包含 $IPP\cdots P$ 帧，没有 B 帧；每个 GOP 中含有 N_{gop} 个帧，每一帧中含有 N_{unit} 个基本单元，每个基本单元中含有 N_{mbunit} 个 MB，则每帧含有的宏块数为

$$N_{\text{mbpic}} = N_{\text{mbunit}} \cdot N_{\text{unit}} \tag{4-87}$$

由于 QP 是按基本单元调整的，因而基本单元越小，输出流的码率波动越小，而重建图像的峰值信噪比(PSNR)值会略低。下面以帧为基本单元进行介绍。

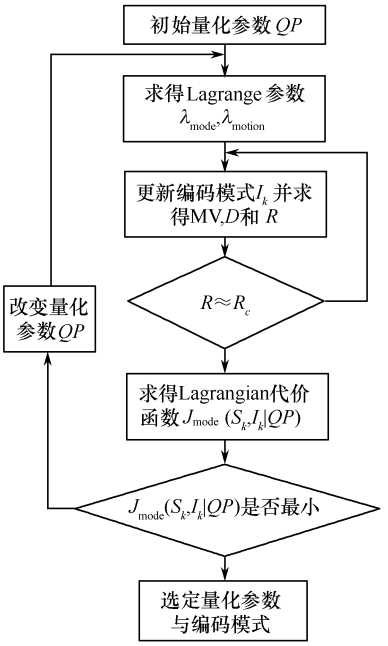


图 4-26　H.264 速率控制和 RDO 的相互作用

1. GOP 级速率控制

在 GOP 的开始，分配给该 GOP 的总比特数按下式计算：

$$T_\gamma(n_{i,o}) = \frac{Rc}{F} \cdot N_{\text{gop}} - \left[\frac{Bs}{8} - Bc(n_{i-1,N_{gop}})\right] \tag{4-88}$$

式中，$n_{i,j}$，($i=1,2,\cdots,j=1,2\cdots,N_{\text{gop}}$)代表第 i 个 GOP 中的第 j 帧，Rc 为可利用的信道带宽，F 为帧率，Bs 为缓存器容量，$Bc(n_{i-1,\text{Ngop}})$为编码完第 $i-1$ 个 GOP 后的缓存器占有量。假设缓存器的初始占有量是 $Bs/8$，上式的分配原则是在编码完每一个 GOP 后，缓存器占有量保持为 $Bs/8$。

在本 GOP 之内，每编码完一帧，其 GOP 目标比特数按下式更新：

$$T_\gamma(n_{i,j}) = T_\gamma(n_{i,j-1}) - A(n_{i,j-1}) \tag{4-89}$$

其中 $A(n_{i,j})$为编码第 i 个 GOP 第 j 帧实际产生的比特数。

在对视频序列的第一个 GOP 进行编码时，该 GOP 的 I 帧和第一个 P 帧采用预先设定的量化参数 QP_0 编码，QP_0 值的选择取决于信道带宽和 GOP 的长度，若信道带宽高或 GOP 较长，QP_0 可以选择得小些。其他 GOP 中的 I 帧和第一个 P 帧，采用下式所示的 QP_{st}进行编码：

$$QP_{\text{st}} = \frac{\text{Sum}_{\text{PQP}}}{N_{\text{p}}} - 1 - \frac{8T_\gamma(n_{i-1,N_{\text{gop}}})}{T_\gamma(n_{i,o})} - \frac{N_{\text{gop}}}{15} \tag{4-90}$$

式中，N_{p} 为 GOP 中 P 帧的总数，Sum_{PQP}为前一个 GOP 中所有 P 帧使用的 QP 值的总和。显然，QP_{st}值也与 GOP 长度和可利用的信道带宽有关。对 QP_{st}值更精细的选择，读者可参见文献[12,13]。

2. 帧级速率控制

(1) 预定义目标缓存器占有量

为了给 GOP 中第二个，以及以后的 P 帧分配目标比特，我们定义一个称为目标缓存器占有量的参数 Tbl，它的初始值为 $Tbl(n_{i,2})=Bc(n_{i,2})$，$Bc(n_{i,2})$ 为编码完第 i 个 GOP 第一个 P 帧后的实际缓存器占有量。后续 P 帧的目标缓存器占有量则为：

$$Tbl(n_{i,j+1}) = Tbl(n_{i,j}) - \frac{Tbl(n_{i,2}) - Bs/8}{N_P - 1} \tag{4-91}$$

从上式不难看出，$Tbl(n_{i,N_{\text{gop}}})\approx Bs/8$。由于实际的缓存器占有量和上式定义的目标占有量可能不同，我们需要进行下面的比特分配以减小二者的差值。

(2) 帧级目标比特分配

根据目标和实际缓存器占有量，以及信道速率和帧率，分配给第 j 帧的目标比特数为：

$$\widetilde{T}(n_{i,j}) = \frac{Rc}{F} + \gamma[Tbl(n_{i,j}) - Bc(n_{i,j})] \tag{4-92}$$

式中，γ 为常数，在无 B 帧时，$\gamma=0.75$；$Bc(n_{i,j})$ 为编码完第 i 个 GOP 第 $(j-1)$ 帧后（即编码第 j 帧之前）的实际缓存器占有量。

在另一方面，根据本 GOP 剩余目标比特数，分配给第 j 帧的比特数为：

$$\hat{T}(n_{i,j}) = \frac{T_\gamma(n_{i,j})}{N_{P,\gamma}} \tag{4-93}$$

式中，$N_{P,\gamma}$ 为剩余的（包括第 j 帧）的 P 帧个数。

综合考虑(4-92)式和(4-93)式，最后分配给第 i 个 GOP 第 j 帧的目标比特数为：

$$T = \beta \cdot \hat{T}(n_{i,j}) + (1-\beta) \cdot \tilde{T}(n_{i,j}) \tag{4-94}$$

式中，β 为常数，且在无 B 帧时，$\beta=0.5$。

(3) 量化参数计算和 RDO 的实现

首先，利用如下的**线性模型**对当前帧的 MAD 值进行估计：

$$\mathrm{MAD}_C = a_1 \cdot \mathrm{MAD}_P + a_2 \tag{4-95}$$

式中，MAD_C 和 MAD_P 分别为当前帧 MAD 的预测值和前一 P 帧 MAD 的实际值，a_1 和 a_2 是线性模型参数，其初始值为 $a_1=1$，$a_2=0$。a_1 和 a_2 在编码过程中不断更新（见(4)）。

然后，根据 4.3.2 节介绍的二次率－失真模型(4-71)式或(4-72)式和分配给该帧的目标比特数 T 计算量化步长 $\hat{Q}$，并将 $\hat{Q}$ 转换成量化参数 $\hat{QP}$（参见表 4-2）。为了使相邻帧的视觉质量比较平滑，我们作如下限定：

$$\widetilde{QP} = \min\{QP_{prev}+2,\quad \max[QP_{prev}-2,\quad \hat{QP}]\} \tag{4-96}$$

式中，QP_{prev} 为前一个 P 帧的量化参数。

最后，量化参数进一步限制在

$$QP = \min\{51,\quad \max[\widetilde{QP},1]\} \tag{4-97}$$

得到 QP 后，则可以对当前帧中的各个宏块进行 4.1.5 节所述的率－失真优化(RDO)和编码。

(4) 模型参数更新

在编完一帧后，(4-95)式所示的线性模型的参数 a_1、a_2 和二次率－失真模型的参数 X_1、X_2 都需要更新。(4-73)式所示的估计 X_1 和 X_2 的方法通常通过递归的方式实现，即

$$X_2 = \frac{n\sum_{k=1}^{n} A_k - (\sum_{k=1}^{n} Q_k^{-1})(\sum_{k=1}^{n} Q_k A_k)}{n\sum_{k=1}^{n} Q_k^{-2} - (\sum_{k=1}^{n} Q_k^{-1})^2} \tag{4-98}$$

$$X_1 = \frac{\sum_{k=1}^{n} Q_k A_k - X_2 Q_k^{-1}}{n} \tag{4-99}$$

式中，n 为用来估计模型参数的数据点对 (A_k, Q_k) 的个数。通常我们选取一个窗口，其中包括 n 个最新编码的帧，A_k 和 Q_k 则是这些帧实际编码的比特数和量化步长。更详细的更新步骤可参见文献[11]。

模型参数 a_1 和 a_2 的更新与 X_1 和 X_2 的更新类似，即使用窗口中的 n 对数据 (MAD_k, MAD_{k-1})，通过最小二乘法来估计。

(5) 缓存器状态更新和跳帧控制

在每编完一帧后，将本帧实际使用比特数加入缓存器占有量 Bc。如果占有量过大，则采取跳帧措施，直到满足下式为止：

$$Bc(n_{i,j+N_{\mathrm{post}}}) < 0.8 \cdot Bs \tag{4-100}$$

式中，N_{post} 是跳过的帧数，且

$$Bc(n_{i,j+l+1}) = Bc(n_{i,j+l}) - \frac{Rc}{F} \qquad 1 \leqslant l \leqslant N_{\text{post}} \tag{4-101}$$

以上我们仅介绍了帧级控制，JVT－G012 还支持基本单元级的速率控制。此外，它还支持 GOP 中包含 B 帧，以及变比特率等情况下的速率控制[12]。

4.3.5 视频缓存证实器

通过对速率控制的讨论可以看出，从一段时间内平均地来看，编码器通过速率控制使其输出码流的速率保持“恒定”；但从某个瞬时来看，其输出码流的速率与其他瞬时并不一定相同。对于恒定速率视频编码器而言，已编码流瞬时速率的变动应不超过一定的范围，在有关的国际标准中这一范围是通过**视频缓存证实器**(Video Buffer Verifier，VBV)来描述的。

VBV 是一个假想解码器的缓存器，如图 4-27(a)所示。假设编码码流从信道上以恒速率 R_c 进入容量为 B 的缓存器，解码器每隔一个帧周期 T_f 从缓存器中取走解码 1 帧图像所需要的数据。由于每一帧图像经压缩编码后产生的比特数不同，因此解码器每次从缓存器中取走的数据量也不同，图 4-27(b)表示出缓存器占有量随时间变化的曲线。在 $t=0$ 时刻，码流开始进入缓存器，其占有量按斜率 R_c 线性上升，到 t_1 时刻解码器取走第 1 帧的数据 d_1，使缓存器占有量降低到 B_1，如此，等等。

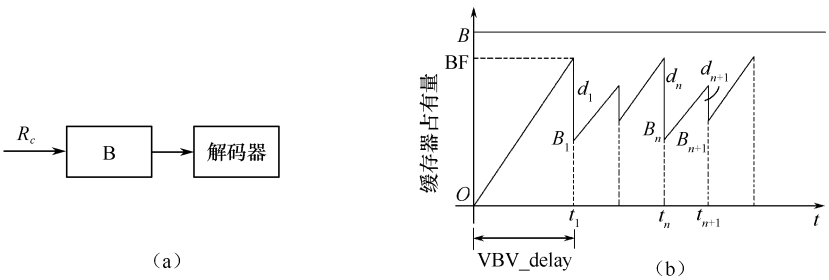


图 4-27　VBV 占有量随时间变化的曲线

设 $t=t_n$ 时刻缓存器剩余量为 B_n，只有在下式成立的条件下，才能满足为第 $n+1$ 帧的解码准备好数据 d_{n+1}的要求(保持缓冲器不变空)：

$$B_n + \frac{Rc}{f_f} \geqslant d_{n+1} \tag{4-102}$$

式中，f_f 为帧率，Rc/f_f 为一个帧周期内进入缓存器的数据。在另一方面，如果在某个时刻，例如 $t=t_{n+1}$，取走的数据量 d_{n+1}很少，缓存器也应该留有足够的空间容纳下一个帧周期内进入缓存器的数据而不溢出，即应保证

$$B_n + \frac{Rc}{f_f} - d_{n+1} + \frac{Rc}{f_f} \leqslant B \tag{4-103}$$

设解码器开始取数据时($t=t_1=$**VBV_delay**)缓存器的占有量为 BF，由(4-102)式和(4-103)式可以推导出 1 帧图像所对应的最大和最小数据量分别为：

$$(d_{n+1})_{\max} = BF + \frac{nRc}{f_f} - \sum_{i=1}^{n} d_i \tag{4-104}$$

$$(d_{n+1})_{\min} = (d_{n+1})_{\max} + \frac{Rc}{f_f} - B \tag{4-105}$$

编码器应该稳定地工作在(4-104)式和(4-105)式所限定的范围内，这也就限定了编码器输出码流瞬时速率的变化范围。同时，参数 R_c、B 和 VBV_delay 被放在规定的数据位上，并复用到已编的码流中，这就使得解码器能够得到这些参数，从而根据上述两式推算出 1 帧编码数据的上限和

下限值，为设计解码器的缓存器提供依据。值得注意的是，解码器缓存器的实际容量并不一定与VBV缓存器的容量 B 相等，它的设计还需要考虑系统的其他要求，如同步、延时等因素。

4.4 变速率视频编码

1. 变速率编码与恒速率编码

我们曾在上一节中对恒定速率编码器的速率控制进行了比较详细的讨论。通过讨论我们知道，I、P 和 B 帧经编码后所产生的比特数有很大差异；而且除了帧类型以外，已编码流的速率还与图像的活动性，即场景的复杂度、运动的快慢和场景切换的多少等有关。假设图像活动性在一定范围内变化，通过仔细的比特分配和输出缓存器的平滑，编码器[见图 4-28(a)]可以向信道输出恒定速率的码流。但是当图像活动性的变化超出一定范围时，则需要根据缓存器的充满程度调节量化器的量化步长，以保证在恒定速率输出的条件下，缓存器不会溢出也会不变空。我们知道，量化步长的变化将改变解码后重建图像的信噪比，图 4-28(a)的右侧给出了恒定码率(CBR)编码的重建图像质量(SNR)变化情况。在某些低延时、低码率的应用(如采用 H.263 的可视电话或会议)中，输出缓存器的容量由于延时要求的限制不能太大，而小的缓存器对码率变化的平滑又是不利的。当图像活动剧烈、编码所产生的数据突发性很强时，可能整个帧的数据都必须舍弃(跳帧)，此时重建图像的信噪比大大降低。

如果信道允许理想的变码率传输，这里的**“理想”**是指信道可以传输任意速率的码流，且解码器具有无限大的缓存器，那么我们可以在编码过程中保持量化步长(即重建图像质量)不变，随着图像活动性变化而产生的变速率码流无需经过缓存器的平滑。此时的编码器称为**“自由(Free)”变速率编码器**。图 4-28(b)给出了自由变速率(VBR)编码器的结构、已编码流速率及重建图像的信噪比的变化情况。除了速率控制部分以外，VBR 的编码算法与 CBR 编码的算法(MC－DPCM、DCT、VLC 等)没有什么不同。

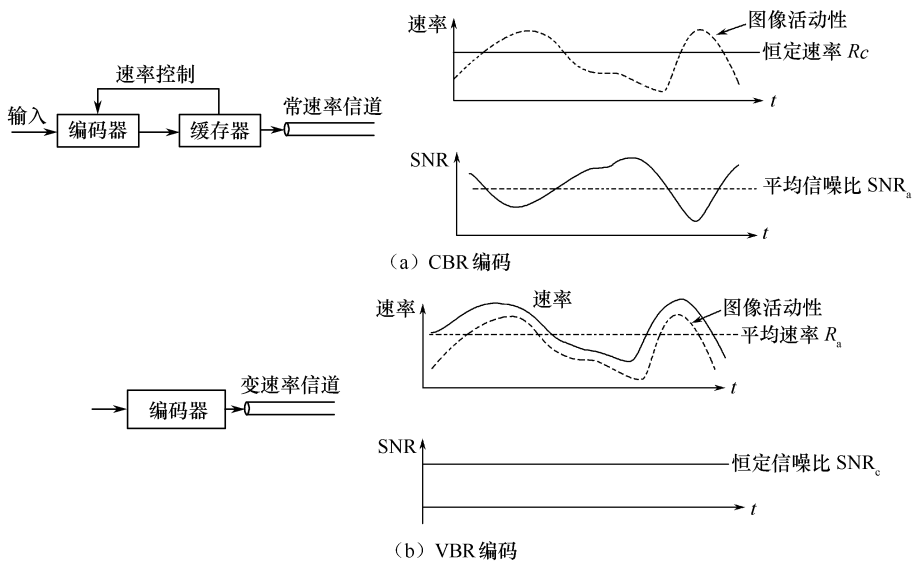


图 4-28 CBR 和 VBR 编码的概念以及速率、信噪比随时间变化的例子

图 4-29 从速率－失真函数的角度表示出 CBR 和自由 VBR 编码的不同。图中的阴影部分简略地描述了速率－失真函数随图像信号特性变化的范围。在 CBR 编码中，由于速率保持为一固定值，失真随着图像特性的变化而在 D_1 到 D_2 的范围内变化[见图(a)]；而在 VBR 编码中，由于失真

固定，速率则随着图像活动性的变化而在 R_1 到 R_2 的范围内变动[见图(b)]。

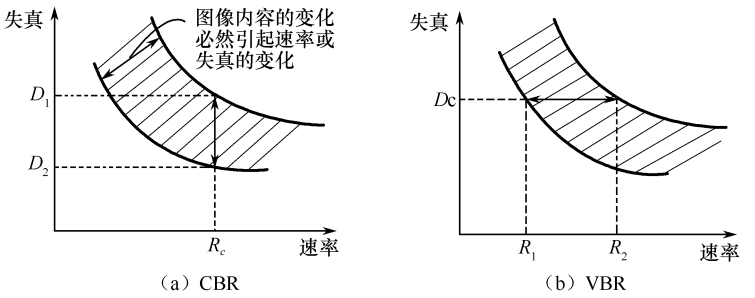


图 4-29　从速率－失真函数的角度对 CBR 和 VBR 编码的比较

在使用相同的编码算法和让恒定速率 R_c 等于 VBR 编码的平均速率 R_a 的条件下，CBR 编码的重建图像的平均信噪比 SNR_a 要低于 VBR 编码重建图像的信噪比 SNR_c，而且重建图像质量的主观评价也较差。在另一方面，当 CBR 和 VBR 编码给出相似的重建图像质量时，VBR 编码给出较低的平均码率和较少的总数据量。例如，对某个 15 分钟的视频序列编码，CBR 编码器给出 6 Mb/s 码流，其总数据量为 675 MB，而 VBR 编码产生的码流的平均速率为 3 Mb/s，总数据量仅为 355 MB[14]。其原因是，如果在某一段时间内视频内容是简单和没有多少运动的，CBR 编码器仍需使用过小的量化步长以产生足够多的比特数来满足 CBR 的要求，从而降低了编码效率。上述两方面的结果表明，VBR 编码无论对于视频数据的传输，还是存储都具有优越性。在 DVD 中采用的就是 VBR 编码。此外，在分组网络中，VBR 编码的码流可以获得更高的统计复用增益。

在自由 VBR 编码器中省去了用于速率平滑的缓存器，从而降低了编码器的延时，也就降低了端到端的传输延时，这对于实时多媒体应用是有利的。

2. 速率约束的 VBR 编码

实际的网络对用户使用的带宽不可能没有任何限制，因此理想的变比特率传输是不实用的。通常的做法是在通信建立时，用户和网络就变比特率传输进行 QoS(峰值速率、平均速率、突发的最大持续时间等)的协商，网络为该连接预留资源。在通信过程中，网络中的流监管对用户数据进行监测，如果其速率超过约定的变化范围，则采取一定的措施加以控制。因此在这种有速率约束的 VBR 编码器中，也存在缓存器和速率控制机制，但缓存器容量比 CBR 编码器中的缓存器要小，而速率控制要兼顾编码和网络两个方面，信源速率的选择应该使编码器达到尽可能高的视频质量，同时信道速率的选择应该使进入网络的流量尽可能地平稳，以达到尽可能高的复用增益。

3. 存储空间约束的 VBR 编码

对于有固定存储空间要求的应用，如 DVD 或 VCD，VBR 编码也是很有用的。此时固定的存储空间相当于要求码流在长时间内的平均速率不超过限定值，但在短时间内码率可以是变化的。由于这种应用不要求实时性，因此通常采用 2 次(Two-Pass)编码的方法进行速率控制。在第 1 次编码时获得整个序列中每一帧的 R-Q 函数(即不同量化步长下产生的比特数)，以便为每一帧选择满足存储空间和解码器缓存容量要求的量化步长；第 2 次编码则使用选择好的量化步长对序列进行实际的编码[15]。

4.5　压缩编码算法性能的评价

数据压缩方法的优劣主要由所能达到的压缩倍数，以及从压缩后的数据所恢复(或称重建)的图像(或声音)质量两个方面来衡量。除此以外，算法的复杂性、延时等也是应该考虑的因素。

1. 数据压缩的倍数

图像压缩的倍数通常由压缩前、后的总数据量之比表示。例如一幅由 720×576 个像素组成的黑白图像，每像素具有 8 b，通过降低分辨率变为 360×288 的图像，又经数据压缩使每个像素平均仅用 0.5 b 表示，则压缩倍数为 64 倍。

衡量压缩倍数的另一种方法是将任何非“压缩算法”产生的效果（如分辨率、帧率的降低等）排除在外，用平均每像素所使用的比特数来表示数据量的大小。可以看到，在上例中压缩前、后的数据量分别为 8 比特/像素和 0.5 比特/像素。

对于视频序列，编码前后的比特率通常是衡量压缩倍数的指标。对于上面的例子，在帧率为 25 帧/s 的情况下，压缩前后的码率分别为 83Mb/s 和 1.3Mb/s。

2. 重建图像质量的客观评价

重建图像的质量通常用重建图像与原图像之间像素值的均方误差（MSE）来衡量。将均方误差作为由数据压缩而产生的噪声能量，可以定义如下 2 种形式的信噪比（SNR）：

$$(\mathrm{SNR})_1 = 10\ \lg\left(\frac{\sigma^2}{\mathrm{MSE}}\right) \quad (\mathrm{dB}) \tag{4-106}$$

$$\mathrm{PSNR} = 10\ \lg\left(\frac{x_{\max}^2}{\mathrm{MSE}}\right) \quad (\mathrm{dB}) \tag{4-107}$$

其中的 MSE 用下式计算：

$$\mathrm{MSE} = \frac{1}{N^2}\sum_{i=1}^{N^2}(x_i - \hat{x}_i)^2 \tag{3-108}$$

上述式子中 x_i 和 $\hat{x}_i$ 分别为原图像和重建图像中对应的像素值，N^2 为 $N\times N$ 图像中的总像素数，σ^2 为输入图像的方差，$x_{\max}$ 为原图像中像素的最大灰度值（对于 8 比特图像，$x_{\max}=255$）。由于(4-107)式定义的简单性，使它成为经常使用的定义信噪比的形式，称为**峰值信噪比**。

3. 视频压缩的率失真性能

两个视频编码器的性能通常通过综合考虑它们压缩后的码率速率和重建图像的信噪比，即它们的率-失真（RD）性能来进行比较。以图 4-30(a)为例，图中的数据点（0 或 x）记录的是两个编码器分别在 4 种不同的量化参数 QP 下，对同一个视频序列编码的结果。拟合 4 个数据点的曲线反映了该编码器的率-失真性能。显然，编码器 2 的性能好于编码器 1，因为在同样的比特率下，它的 PSNR 值要高。

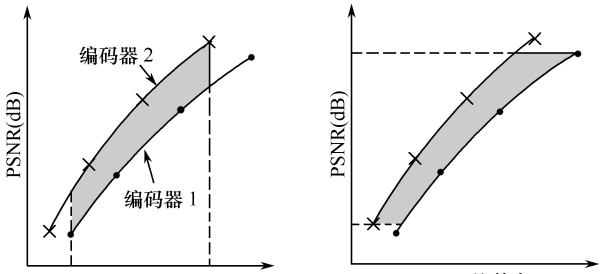


图 4-30　视频编码器性能比较

RD 曲线的比较虽然直观，但不能给出定量的结论。国际电联的视频编码专家组（VCEG）建议了一种计算两条曲线的平均信噪比差（BD-PSNR）和平均比特率差（BD-rate）的方法[16]，并提供了相应的工具[17]。该方法首先将比特率用 $\log_{10}$ 表示（以 dB 为单位的 PSNR 已经取了对数）；对数坐

标下的 4 个数据点的 RD 曲线，可以用三阶多项式来拟合，即

$$\mathrm{PSNR}=a+b\cdot B+c\cdot B^2+d\cdot B^3 \tag{4-109}$$

或者

$$B=a+b\cdot \mathrm{PSNR}+c\cdot \mathrm{PSNR}^2+d\cdot \mathrm{PSNR}^3$$

式中 B 为比特率。对两条曲线的拟合函数分别积分并求差值（图中灰色区域），再除以积分区间，就得到 BD-PSNR（或 BD-rate）。为了不必对测试数据进行外插，计算平均信噪比差和平均比特率差所使用的积分区间分别如图 4-30(a)和(b)中虚线所示。

4. 重建图像的主观评价

由于信噪比并不能完全反映人对图像质量的主观感觉，国际电联无线电组在标准 BT 500-11 中规定了在严格的观测条件（图像尺寸、对比度、亮度、观察距离、照明等）下，对一组标准图像压缩前后的质量进行对比的主观评定标准。具体的做法是，由若干人（分专业人员组和普通人员组）对所观测的重建图像的质量（很好、好、尚可、不好、坏），或对重建图像中噪声的明显程度（不可见、可见、明显、令人烦恼、无法看）按 5 级分评定，然后按下式计算**平均评价分数**（Mean Opinion Score，MOS）：

$$\mathrm{MOS}=\sum_{i=1}^{k} n_i c_i \Big/ \sum_{i=1}^{k} n_i \tag{4-110}$$

式中，k 为类别数，即有几个分数等级，c_i 为分数，n_i 为该类别的人数。如果将 MOS 看成是一个随机变量的均值，那么相对于此均值的方差反比于参加测试的总人数 $N=\sum_{i=1}^{k} n_i$。这说明测试规模需要足够大，才能得到较稳定的评价。

对于音频数据压缩算法的质量评价也与此类似，可以用信噪比、各种形式的加权信噪比，以及主观评定的 MOS 等来度量。

粗略地说，对于峰值信噪比高于 40 dB 的图像，通常主观感觉是“很好”的，而 20 dB 以下的图像往往是“不好”的。但是主观感觉还受到许多其他因素的影响，例如对于一个背景模糊、而前景较清晰的图像，人的主观评价可能比对一个整体信噪比略高、而背景和前景不予区分的图像要高。由于主观评测费时、费力，而且不能将其集成进系统的质量反馈控制环路中去，因此寻求能够准确地反映主观评价的客观测试方法，已成为一个吸引人的研究领域，ITU-T 的视频质量专家组（VQEG）正在努力建立相关的标准[18]。这方面的成果及其在图像和视频编码以及多媒体业务的**用户体验质量**（见 7.2.3 节）等方面的应用值得关注。

4.6 图像和视频压缩编码的国际标准

4.6.1 静止图像压缩编码标准 JPEG/JPEG2000

静止图像压缩编码标准 JPEG（ISO/IEC 10918）是由 ISO 和 ITU-T 组织的联合摄影专家组（Joint Photographic Expert Group）为单帧彩色图像的压缩编码而制定的，图像尺寸可以在 1～655 行/帧、1～65 535 像素/行的范围之内。采用这一标准可以将每像素 24 比特的彩色图像压缩至每像素 1～2 比特仍具有很好的质量。

JPEG 定义了 4 种模式：(1)**顺序(Sequential)方式**，其基本算法是将图像分成 8×8 的块，然后进行 DCT、量化和熵编码（霍夫曼码或算术编码）；(2)**渐近(Progressive)方式**，所采用的算法与方式(1)相类似，不同的是，先传送部分 DCT 系数信息（例如低频带系数，或所有系数的近似值），使收端尽快获得一个“粗略”的图像，然后再将剩余频带的系数（或所有系数的低比特位数据）渐次传送，最终形成清晰的图像；(3)**无损方式**，采用一维或者二维的空间域 DPCM 和熵编码。由于输入

图像已经是数字化的，经空间域的 DPCM 之后，预测误差值也是一个离散量，因此可以不再量化而实现无损编码；(4)**分层(Hierachical)方式**，在此方式中，首先将输入图像的分辨率逐层降低，形成一系列分辨率递减的图像。先对分辨率最低的底层图像进行编码，然后，将经过内插的低层图像作为上一层图像的预测值，再对预测误差进行编码，以此类推，直至顶层。

当该标准用于序列图像的编码时，称为 **Motion JPEG**(MJPEG)。此时，帧间图像信息的相关性未被利用，只进行了帧内编码。

联合摄影专家组于 2000 年年底又制定了一个新的静止图像压缩标准 JPEG2000(ISO/IEC 15444-1 或 ITU-T Rec. T. 800)，它比 JPEG 有更高的编码效率。JPEG2000 与 JPEG 的主要不同之处有如下几个方面。(1)**高效的编码方法**。JPEG2000 采用小波变换和高效的数据组织方式 EBCOT(见 3.8 节)代替 8×8DCT，并采用算术编码，使得在同样压缩比下编码性能比 JPEG 提高约 2 dB；(2)**ROI 编码**。在有些应用中，图像的某个区域比其他区域(背景)具有更高的重要性，这个重要区域常称为**感兴趣区 ROI**(Region Of Interest)。JPEG2000 支持对 ROI 进行比背景质量更高的编码；(3)**可伸缩性编码**。JPEG2000 支持类似于分层方式的空间分辨率可伸缩性和信噪比可伸缩性编码。后者意味着可以在已编码流的不同点截断，以获得信噪比不同的重建图像。

4.6.2 视听会议压缩编码标准 H.261

H.261 是由 ITU-T 为在窄带综合业务数字网(N-ISDN)上开展速率为 $p*64$ kb/s 的双向声像业务(可视电话、会议)而制定的，其中 $p=1\sim30$，H.261 也称为 $p*64$ 标准。

H.261 只对 **CIF**(Common Intermediate Format)和 QCIF(Quarter CIF)两种图像格式进行处理。由于世界各国采用的电视制式不同(如 PAL、NTSC、SECAM)，所规定的图像扫描格式(决定电视图像分辨力的参数)也不同，要在这些国家之间建立可视电话或会议业务，无法直接采用电视所规定的格式，而必须统一到一个公共图像格式上来，这也是 CIF 这个名称的由来。CIF 和 QCIF 的亮度信号分辨率分别为 352×288 和 176×144，色差信号 C_r 和 C_b 的水平和垂直分辨率均为亮度信号的二分之一，即分别为 176×144 和 88×72。图像帧率最高为 29.97 帧/秒。在信道速率较低时，帧率可以降至 10 帧/秒左右。

H.261 压缩编码由具有运动补偿的帧间预测、块 DCT 和霍夫曼编码组成。由于该标准用于实时业务，希望编/解码延时尽可能小，所以只利用前一帧作参考帧进行前向预测。除初始帧为 I 帧外，后续帧一般为 P 帧。为了防止信道误码产生的差错经预测编码而累积传播，在每 132 帧之内，在每个宏块位置上至少要进行 1 次帧内编码(即**逐步刷新**)。此外，当某个宏块预测误差太大时，也可以做帧内编码。编、解码器的复杂程度相当(或称为对称)是 H.261 标准的一个特点，这是因为会话的双方都需要同样的编码器和解码器的缘故。

4.6.3 数字声像存储压缩编码标准 MPEG－1

该标准(ISO/IEC 11172)由 ISO 活动图像专家组(Moving Picture Expert Group)为速率为1～1.5 Mb/s 能够提供录像质量(VHS)的数字声像信息的存储而制定，共分为视频编码、声音编码和声像同步与复用(系统)3 个部分。

MPEG－1 可以处理的图像格式为 **SIF**(Source Input Format)格式，即亮度信号为 352×240 像素×30 帧/秒(NTSC)或 352×288 像素×25 帧/秒(PAL)。此时在 4∶2∶0 模式下，视频信号压缩后的码率约为 1.2 Mb/s；再加上压缩以后、具有 CD 质量的双声道立体声伴音，总速率约为 1.4 Mb/s。

MPEG 1 采用 4.1.1 节中介绍的典型编码算法。由于针对数字存储的应用而制定，因此它的编、解码器是不对称的，位于存储中心的编码器比位于用户端的解码器要复杂得多。此外，为了支持对存储在数字存储介质上的已编码流进行随机的存取(例如快进、快倒等操作)，MPEG－1 码流

通常具有 GOP 结构。

4.6.4 通用视频及伴音压缩编码标准 MPEG－2(H.262)

MPEG－2(ISO/IEC 13818)由 ISO 的活动图像专家组和 ITU-T 于 1994 年共同制定,在 ITU-T 的协议系列中,被称为 H.262。制定 MPEG－2 的初衷是得到一个针对广播电视质量(CCIR 601 格式)的视频信号的压缩编码标准,但实际上最后得到的是一个通用的标准,它能在很宽范围内对不同分辨率和要求不同输出比特率的视频信号有效地进行编码。表 4-3 给出了其中的一些例子。

表 4-3 MPEG－2 的几种应用

应　　用	分　辨　率	压缩后的码率
VHS 质量电视	352×288 25 Hz	1.5 Mb/s
广播质量电视	720×576 25 Hz	4～10 Mb/s
HDTV	1 920×1080 30 Hz	30～40 Mb/s

像 MPEG－1 一样,MPEG－2 也分为系统、视频和音频 3 个部分。其中系统部分在 ITU-T 协议中称为 H.222.0,我们将在 8.2.2 节中讨论。

MPEG－2 的编码方式与码流结构与 MPEG－1 相似。与 MPEG－1 相比,MPEG－2 主要增加了下述几项功能。(1)**处理隔行扫描的视频信号的能力**。增加了场图像的场间预测、帧图像的场间预测等 4 种预测模式,以及图 4-5(b)所示的 2D 到 1D 的扫描顺序等对隔行扫描图像更有效的编码方式;(2)**更高的色信号取样模式**。除了 4∶2∶0 外,还支持 4∶2∶2 和 4∶4∶4 模式;(3)**可伸缩的视频编码方式**。所谓可伸缩性视频编码是指一次编码所产生的码流具有下述特性:对码流的一部分进行解码可以获得完整的重建图像,但其质量要比对全部码流解码获得的图像质量(分辨率、帧率、或信噪比等)要低。MPEG－2 所支持的可伸缩性视频编码方式有空间可伸缩性、时间可伸缩性、信噪比可伸缩性和数据分割 4 种,我们将在 10.2 节中详细介绍。

为了适应不同应用的需要,MPEG－2 用**范畴**(Profile)以及范畴中的一个或几个**层次**(level)来描述不同的编码参数集。每个范畴是前一个的合集(Superset),规定了一组可用的编码工具或算法;层次则规定了对一些关键参数的限制,例如空间和时间分辨率的上限。MPEG－2 定义的范畴和层次分别见表 4-4 和表 4-5,其中 Main/Main 的范畴/层次组合(MP@ML)是 MPEG2 的主要应用形式,应用于标准清晰度的数字电视。

表 4-4 MPEG－2 范畴

范　　畴	特　　征
简单(Simple)	4∶2∶0 取样,仅用 I/P 帧,无可伸缩性
主(Main)	以上参数,加上 B 帧
信噪比(SNR)	以上参数,加上 SNR 可伸缩性
空间(Spatial)	以上参数,加上空间可伸缩性
高(High)	以上参数,改为 4∶2∶2 取样

表 4-5 MPEG－2 层次

层　　次	最高分辨率
低(Low)	352×288　亮度信号样值,30 Hz
中(Main)	720×576　亮度信号样值,30 Hz
高(High)－1440	1 440×1 152 亮度信号样值,60 Hz
高(High)	1 920×1 152 亮度信号样值,60 Hz

4.6.5 低比特率视听会议压缩编码标准 H.263

H.263 是 ITU-T 制定的低比特率(64 kb/s 及以下)视频信号压缩标准,主要应用于可视电话和视频会议。在 H.263 的发展中,先后出现过两个改进版本,H.263＋和 H.263＋＋,这些改进的内容目前已经作为不同的 Annex 补充进标准的文本。表 4-6 给出了 H.263 所支持的图像格式。除此之外,它也支持用户自定义的图像格式。

表 4-6　H.263 所支持的图像格式

图像格式	亮度样值/行	亮度行数/帧
Sub-QCIF	128	96
QCIF	176	144
CIF	352	288
4 CIF	704	576
16 CIF	1 408	1 152

H.263 是在 H.261 的基础上加以改进而形成的，它的主要改进如下所述。(1)**高效率的编码模式。**增加了若干编码模式，以提高编码效率。这包括半像素精度的运动补偿、不受限的运动矢量、8×8 块的帧间预测、DCT 系数的空间预测和基于句法(Syntax-based)的算术编码等；(2)**PB 帧模式。**为了保证电话和会议应用的实时性要求，H.263 不使用附加延时较大的双向预测 B 帧，而采用了 P 帧和 B 帧作为一个单元来处理的方式，即将 P 帧和由该帧与上一个 P 帧所共同预测的 B 帧一起进行编码，称为 PB 模式；(3)**抗误码措施。**为了改善在高噪声信道上视频传输的质量，H.263 在编码上增加了一些抗误码模式。例如，参考图像选择、错误跟踪、独立分段解码等，我们将在 9.4 节中给予详细讨论。

4.6.6　通用音视频对象压缩编码标准 MPEG－4

ISO MPEG－4(ISO/IEC 14496)的初衷是制定一个通用的低码率(64 kb/s 以下)标准，并打算采用第二代压缩编码算法，以有效地支持甚低码率(Very Low Bit Rate)的应用。但是，由于压缩编码的第二代算法，在预定的时间(1997 年)内还不够成熟，因而其目标后来转向支持已有的标准尚未全面支持的那些应用，例如，与其他媒体数据(如计算机产生的图形、图像)的集成和交互式多媒体服务等。MPEG-4 通过**工具**、**对象**和**范畴**来定义编码功能。范畴定义编解码器能处理的对象；对象定义使用某种或多种工具编码的视频单元；工具定义一组编码方法。例如，“简单”范畴所处理的视频对象是由工具的一组子集所编码的矩形视频帧序列。在 MPEG－4 中对一些关键参数的限制也由**层次**来定义。

与其他标准不同，MPEG－4 并不是一个单纯的视音频压缩标准，它还能够描述由若干个对象所构成的多媒体场景，这些对象可以是视频、音频、计算机产生的 2D 或 3D 物体、动画、电子音乐、文字或图形等。从自然界摄取的视频图像序列可以以矩形帧为单元，也可以以任意形状的对象为单元进行编码。图 4-31 给出了多媒体场景描述和 MPEG－4 编码码流之间的关系，图中表示由一个背景和 2 个前景视频图像的叠加所组成场景。MPEG－4 采用**场景二进格式 BIFS**(Binary format for scenes)来描述多媒体场景，即用 BIFS 来定义构成场景的对象之间的空间和时间关系。这些关系通常以树的结构表示。如图所示，每一个对象对应于树上的一个叶子。有关某个对象的信息由**对象描述符 OD**(Object Descriptor)表示。通过 OD 可以找到对应于该对象的媒体流，媒体流中包含对象的运动、纹理和形状信息。场景描述符(BIFS)流、对象描述符流和媒体流复接成一个流输出。接收端首先通过一个初始 OD 找到 BIFS 流和 OD 流，然后通过这两个流找到构成场景的各个媒体流。在 MPEG－4 中，上述接入过程通过一组 API 函数来实现，该 API 称为 **DMIF**(Delivery multimedia integration framework)应用接口。

如上所述，MPEG－4 场景是由对象构成的，在 OD 流中还可以插入“加入”、“删除”和“更新”等 OD 命令，这使得用户可以直接对“对象层”进行存取操作，例如，有选择地只对其中的几个对象解码和显示、对其中某个对象进行放缩、移位和旋转等操作(编辑)以后再解码显示、或增加原图像中没有的新对象等，并能很容易地进行自然图像和计算机图形学的方法生成图像的综合。

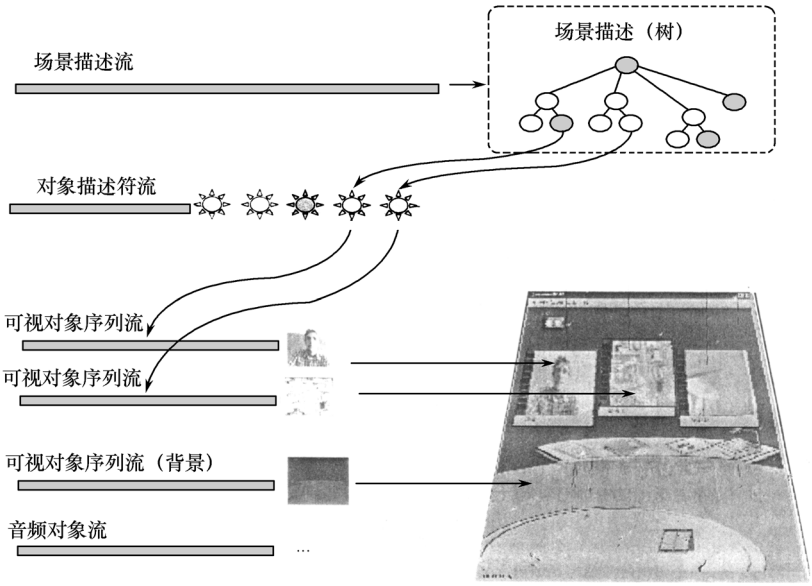


图 4-31　MPEG-4 场景构成

在可视对象的编码方面，MPEG4 具有以下特点。(1)**高效率和强鲁棒性的编码。**低码率的核心编码以 H.263 编码为基础，借鉴并改进了其中高效的编码工具，例如 8×8 的帧间预测、不受限的运动矢量和 DCT 系数的预测编码，以及允许 1/4 像素精度运动补偿和全局运动补偿等。同时，还采取重复传送包头信息、可逆变长编码、数据分割和动态分辨率转换等措施（见 9.4 节），提高已编码流在噪声信道上传输的鲁棒性。正是由于上述原因，以矩形帧为基本单元的 MPEG－4 编码在低码率和高噪声环境下得到了广泛的应用。(2)**静止背景编码。**在许多情况下，背景在整个视频序列中没有太大的变化，此时可以将它作为一个整体，称为 **Static Sprite**，进行编码和传输。通常将大于屏幕可见区域的完整背景压缩后先传送给接收端，然后接收端根据后续接收到的参数对背景进行一定的平移和卷曲，并将适当的部分呈现在屏幕上来模拟摄像机位置和角度变化产生的效果。这种方式大大提高了背景的压缩效率。(3)**可伸缩性编码。**支持空间可伸缩性、时间可伸缩性和信噪比可伸缩性编码。在信噪比可伸缩性中，它支持**细粒度可伸缩性 FGS**(Fine Granular Scalability)，为传输速率的调整提供了极大的灵活性，我们将在 10.2 节中加以讨论。(4)**小波纹理编码。**允许使用小波变换进行图像的编码。这里的图像可以是覆盖在 2D/3D 动画框架上的表面纹理，也可以是一个任意形状对象的纹理部分，或一个矩形视频帧。(5)**动画对象的编码。**支持 2D/3D 网格(mesh)对象的编码。2D 网格由覆盖对象平面的三角形组成，而 3D 网格由 3D 空间中的一组多边形组成。这些网格表面覆盖上适当的纹理图像就构成了不同的动画对象。MPEG－4 还提供一组人脸和身体动画工具，为人脸和身体建模和编码，这在极低码率的视频编码中有潜在的应用价值。

4.6.7　视频压缩编码标准 H.264/AVC

由 ISO MPEG 和 ITU-T 的视频编码专家组 VCEG 组成的联合视频组 JVT(Joint Video Team)于 2003 年制定了一个视频压缩编码标准，称为 ITU-T H.264 建议或 MPEG-4 的第 10 部分，简称 H.264/AVC(Advanced Video Coding)。H.264 不仅具有高压缩比(在相同视频质量下，比 MPEG2 的比特率降低 50%)，而且在恶劣的网络传输条件下，具有较高的抗误码性能。目前，H.264 已成为多媒体领域最为广泛使用的视频编码标准。

H.264 支持表 4-7 所示的三个范畴。

表 4-7　H.264 的几种应用

范畴(Profile)	应　　用
基本(Baseline)	视频会话,如可视电话、远程医疗、远程教育、会议电视等
扩展(Extended)	网络的视频流,如视频点播、IPTV 等
主要(Main)	消费电子应用,如数字电视广播、数字电视存储等

与 H.263 和 MPEG－4 相比,H.264 主要做了如下改进:

(1) **帧内预测编码。**对于有大量细节的图像区域,采用 4×4 亮度像素子块的帧内预测编码,共有 9 种预测模式;对于图像平坦区域,采用 16×16 亮度块的帧内预测编码,共有 4 种预测模式;而对于色度块,则采用 8×8 的帧内预测编码,共有 4 种预测模式;(2)**帧间预测多模式选择。**帧间预测可以采用图 4-13 所示的各种子块尺寸;同时可以从前后多个(最多 15 个)已编码的图像中选择当前帧的参考帧。其 B 帧不仅可由传统的一前一后两参考帧预测,还可以由两个前向帧或两个后向帧进行预测;(3)SP/SI **帧。**定义了两种新的帧类型:SP 帧和 SI 帧。SP 帧也采用具有运动补偿的帧间预测编码,与一般 P 帧不同的是,SP 帧可以参照不同的参考帧重构出相同的帧,因此在 SP 帧处可以像在 I 帧那样进行两个速率或帧率不同的流之间的切换、拼接、随机接入、快进快退及错误恢复等(见 8.6.4 节)。由于 SP 帧的码率开销远比 I 帧要低,因此降低了上述应用中的带宽要求。SI 帧是基于帧内预测的编码技术,其功能和实现原理与 SP 帧类似,主要用于不同内容(节目)的流之间的切换;(4)**低复杂度变换与量化。**采用了 4.1.6 节介绍的 4×4 变换替代先前各标准中的 8×8 离散余弦变换,并将变换与量化相结合,降低了计算复杂度;(5)**对传输错误的鲁棒性和对不同网络的适应性。**在视频编码和网络传输层之间定义了一个**网络抽象层 NAL**(Network Abstraction Layer),将视频码流封装进 NAL 单元,可以更灵活地与不同的网络相适配(见 9.4.2 节)。同时,支持**灵活宏块排序 FMO**(Flexible Macroblock Ordering)、任意条带排序和数据分割等方式,增强了码流抵抗误码和丢包的鲁棒性(见 9.4.2 节)。

4.6.8　视频压缩编码的国家标准——AVS

AVS(Audio Video coding Standard)是我国具有自主知识产权的视频编码国家标准,它也分为音频、视频和系统 3 个部分。其中视频标准定义了"基准"范畴(Profile)和 4 个层次(Level),支持的最大图像分辨率从 720×576 到 1 920×1 080,最大比特率从 10 Mb/s 到30 Mb/s。

AVS 也采用典型的混合编码框架,包括帧内和帧间预测、变换、量化和熵编码等。它与其他国际标准不同之处在于提出了一批具体的优化技术,例如,采用 16 位精度加法和移位即可完成的 8×8整数变换与量化、多种块大小的运动补偿和多参考帧预测、二维熵编码,以及不同的内插滤波器和环路(去块效应)滤波器等,在稍低的复杂度下实现了与相关国际标准相当的技术性能。

4.6.9　正在制定的视频压缩编码标准——HEVC

随着近几年来技术的迅速发展,电视已从模拟转向了数字化,高清晰度的电视和视频点播,以及兰光 DVD 逐渐普及;标准清晰度、隔行扫描的 CRT 显示器已被面积和分辨率不断增大的平板显示器所代替;许多多媒体业务,例如视频会议,已从使用专用通信线路和会议室系统转向使用因特网和家用或移动终端设备。为了适应新形势下对视频压缩编码的要求,MPEG 和 ITU VCEG 于 2010 年 1 月建立了联合视频编码组 JCT-VC(Joint Collaborative Team on Video Coding),并着手制定新的视频压缩国际标准。该标准目前被称为 HEVC(High efficiency video coding),或者 H.265。

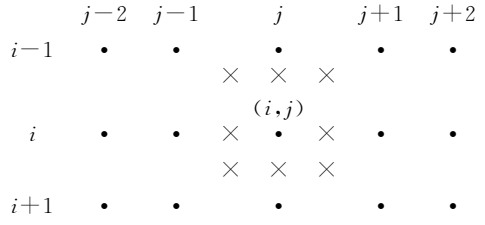
新标准的目标是,处理更高分辨率和更大尺寸的图像(最大 8K×4K),在相同视频质量下,将

压缩后的比特率比 H.264 降低 50%，且算法有相对低的计算复杂度和延时。

新标准仍以混合编码框架为基础，但引进了更多的优化技术。现在可以看到的趋势是：(1)为了适应分辨率高于 HDTV(≤1920×1080)、甚至于 Ultra HDTV(最大 7600×4320)图像在时间和空间上变化的剧烈性，编码方式和编码参数大多采取自适应的模式。例如，编码的基本单元不再是固定大小的宏块，而是尺寸更大的、不固定大小的块；进行运动估值和变换编码的块的大小和形状也有更多的选择性；亚像素内插滤波器和运动矢量的精度(1/2,1/4,1/8,1/12)不再采用固定的，而可以根据残差最小的原则动态地选择；变换编码也可以根据图像内容而采用不同的形式等；(2)为了使参考图像更接近于原图像，在预测环路中除了去块效应滤波器外，又增加了一个自适应环路滤波器(Adaptive Loop Filter，ALF)。(3)由于自适应处理往往需要在编码过程中对视频作多次扫描(Multi-pass)，因此算法具有大规模并行处理的结构就很值得考虑。

习 题 四

1. 在图 4-1 所示的典型编码器中
 (1) 经上述编码器压缩的图像，在解压缩之后会出现“块效应”，产生“块效应”的原因是什么？
 (2) 阐述在 DCT 变换之后进行之字形扫描的目的；对隔行扫描的帧图像块为何要采用图 4-5(b)的扫描方式。
 (3) 为什么要进行速率控制？保持编码器输出速率恒定，所付出的代价是什么？
2. 为什么对帧内编码宏块和对帧间编码的宏块要采用不同的量化矩阵(如图 4-4 所示)？通常在熵编码之前对这两种宏块的 DC 分量的处理是否相同？为什么？
3. 运动矢量的估值可以达到半个像素的精度。所谓**半像素位移矢量估值**是首先在原始分辨率的图像上搜索到 SAD 最小的位置(i,j)(见下图)，然后通过内插得到(i,j)周围的 8 个半像素点(见图中×的位置)的灰度值，在这 8 个点再加上(i,j)点本身共 9 个点中进行搜索，找到最终的匹配点。若 k 帧中待预测块的中心坐标为(168,152)，$k-1$ 帧中(i,j)点的坐标为 $i=147$，$j=163$，匹配块中心位于(i,j)点右上角的半像素点，写出半像素点精度的位移矢量。



4. 设原始图像块如下图所示，按下列步骤进行 DCT 变换编码：

$$\begin{bmatrix} 139 & 144 & 149 & 153 & 155 & 155 & 155 & 155 \\ 144 & 151 & 153 & 156 & 159 & 156 & 156 & 156 \\ 150 & 155 & 160 & 163 & 158 & 156 & 156 & 156 \\ 159 & 161 & 162 & 160 & 160 & 159 & 159 & 159 \\ 159 & 160 & 161 & 162 & 162 & 155 & 155 & 155 \\ 161 & 161 & 161 & 161 & 160 & 157 & 157 & 157 \\ 162 & 162 & 161 & 163 & 162 & 157 & 157 & 157 \\ 162 & 162 & 161 & 161 & 163 & 158 & 158 & 158 \end{bmatrix}$$

 (1) 将图像块中各像素值减去均值 128；
 (2) 对由(1)所得到的矩阵进行二维 DCT；
 (3) 采用图 4-4(a)的量化矩阵量化；
 (4) 采用之字形扫描得到一个一维序列(用 EOB 表示序列结束)；
 (5) 求重建的图像灰度矩阵。
5. 如下的函数中，哪些可以作为描述 8×8 块内的亮度变化剧烈程度的空间活动度函数？对 A_F 值较大的块，量化时所分配的比特数应该多，还是少？解释原因。

(a) $A_F=\sum_{i=1}^{64}|p_i|^2$　　(b) $A_F=\sum_{i=2}^{64}|p_i|^2$　　(c) $A_F=\sum_{i=1}^{64}|p_i-\bar{p}|^2$

(d) $A_F=\sum_{i=1}^{64}|y_i|^2$　　(e) $A_F=\sum_{i=2}^{64}|y_i|^2$　　(f) $A_F=\sum_{i=1}^{64}|y_i-\bar{y}|^2$

其中 p_i 为块内的像素值，y_i 为该块对应的 DCT 系数矩阵中的元素值，$\bar{p}$ 和 $\bar{y}$ 分别为 p_i 和 y_i 的均值。

6. 一个逐行扫描的帧内编码块，经 DCT 和量化之后，得到如图示的结果(未写数字的位置其值为 0)，

(1) 将图示的矩阵排成一维的数据序列；

(2) 写出该序列经游程编码后的结果；

(3) 按照表 4-1 对该序列进行霍夫曼编码。

4	−1			1			
	−3						
2							
−1							

7. 在 JPEG 标准中，经 DCT 和量化后的系数在变成一维序列(*run*，*level*)之后，又将 *level* 分解成(*sss*，*Value*)，其中 *sss* 表示将 *level* 编码所需的比特数，*Value* 为该值的二进制数值(负值用取反表示)；然后将(*run*/*sss*)作为符号，构成如下的霍夫曼码表。每一个(*run*，*level*)最后表示为(*run*/*sss*)霍夫曼码＋*value* 二进制码。根据上述原则和下表写出(0,6)(0,7)(3,3)(0,−1)(0,0)经编码后的码流。

run/*sss*	码字
0/3	100
3/2	111110111
0/1	00
0/0	1010

8. 在上题中只对(*run*/*sss*)进行霍夫曼编码，对 *value* 则采用二进制码；相类似地，4.1.1 节中介绍的 H.263/MPEG−4 DCT 系数只对出现频率高的(*run*，*level*)组合进行霍夫曼编码，出现频率低的则采用二进制编码，请解释为什么不在所有情况下都用霍夫曼码？

9. 若在一个有 6 帧图像的 *GOP* 中，各帧的比例为 $I:P:B=1:2:3$。画出该 *GOP* 序列，用箭头标明 *P* 帧和 *B* 帧的参考帧，并给出各帧输入、编码、传输、解码和显示的顺序。说明编码延时和解码延时各为多少，以及延时与 *B* 帧数目的关系。

10. 求下列矩阵的 4×4 变换(4.1.6 节)和 DCT 变换，并比较它们的差别。

8	9	8	10
11	5	12	4
19	10	11	4
1	6	15	7

11. 在进行基于对象的预测编码中，对于边缘块中的非对象区域可以(1)填充 0；(2)填充 $2^{n-1}-1$，其中 *n* 为每像素的比特数；(3)进行图 4-22 所示的扩展填充。为什么要进行填充？填充是否会引起重建图像的失真？请对这三种方法的性能进行比较。

12. 为什么在 TM5 速率控制算法中，不考虑“跳帧”？

13. 假设在一个低延时和低码率的应用中，我们在速率控制中采用 TMN8 的帧级控制为 *P* 帧分配目标比特，但由于在码率很低的情况下，每帧编码使用同一 *QP* 比每个宏块改变一次 *QP* 有更好的效果，因此我们将不采用 TMN8 的宏块级控制，而是在帧级直接使用二次率−失真模型确定量化参数。写出此速率控制算法的主要步骤(包括模型参数的更新)。

14. 请将一幅含有丰富细节的 16CIF 图像通过简单丢弃样值的方法得到一幅 QCIF 图像，观察并指出是否存在混

叠失真。设计并编程实现一个抗混叠滤波器，获得无可见混叠失真的 QCIF 图像。

参 考 文 献

[1] B. Girod, "Motion-compensating prediction with fractional-pel accuracy," IEEE Trans. on Communications, Vol, 41, No. 4, 1993, 604-611

[2] T. Wedi and H. G. Musmann, "Motion and aliasing compensated prediction for hybrid video coding," IEEE Trans, on CSVT, Vol. 13, NO. 7, 2003, 577-586

[3] T. Wiegand and B. Girod, "Lagrange multiplier selection in hybrid video coding control," ICIP, 2001, 542-545

[4] F. Pan, el al. , "Fast mode decision for intra prediction," JVT-G013, 2003

[5] K. P. Lin, el al. , "Fast inter mode selection," JVT-1020, 2003

[6] B. Jeon and J. Lee, "Fast mode decision for H. 264," JVT-J033, 2003

[7] J, Lee, el al. , "Fast mode decision for B slice," JVT-K021, 2004

[8] H. S. Malvar, el al. , "Low-complexity transform and quantization in H. 264/AVC," IEEE Trans. CSVT, Vol. 13, No. 7, 2003, 597-603

[9] I. E. G. Richardson, H. 264 and MPEG－4 Video Compression, John Wiley & Sons, 2003

[10] J. Ribas-Corbera and S. Loi, "Rate control in DCT video coding for low-delay communications," IEEE Trans. on CSVT, Vol. 9, No. 1, 1999, 172-185

[11] H. J. Lee, el al. , "Scalable rate control for MPEG-4 video," IEEE Trans. on CSVT, Vol. 10, No. 6, 2000, 878-894

[12] Z. G. Li, el al. , "Adaptive basic unit layer rate control for JVT," JVT-G012, 2003

[13] Z. G. Li, el al. , "Adaptive rate control for H. 264," ICIP, 2004, 745-748

[14] S. Gringeri, el al. , "Transmission of MPEG-2 video stream over ATM," IEEE Multimedia, 1998, 58-71

[15] Y. Yu, el al. , "A novel two-pass VBR coding algorithm for fixed-size storage application," IEEE Trans. on CSVT, Vol. 11, No. 3, 2001, 345-356

[16] G. Bjontegaard, "Calculation of Average PSNR Difference between RD Curve," ITUT-T SG16/Q6, Doc. VCEG-M33, Apr. 2001.

[17] K. Senzaki, et al, "BD-PSNR/rate computation tool for five data points", DOC. JCTVC-Boss, 2010.

[18] http://www. vqeg. org

第 5 章　音频数据的压缩编码

5.1　概述

音频压缩编码研究的基本问题是，在给定编码速率的条件下，如何能够得到尽量好的重建语音质量，同时应该尽量减少编解码延时以及 9 算法的复杂程度；或者说在给定编码质量、编解码延时及算法复杂度的条件下，如何降低语音所需的比特率。这 4 个因素之间有着密切的联系，在不同的应用中对各方面的侧重要求也有所不同。

音频信号通常可以分为话音（人的说话声音）和一般声音（如音乐）两类。对于语音编码有很多种划分方法，例如：

按照编码速率来分，可以将语音编码分为 5 类：高速率（32 kb/s）以上、中高速率（16～32 kb/s）、中速率（4.8～16 kb/s）、低速率（1.2～4.8 kb/s）和极低低速率（1.2 kb/s 以下）。

按照被编码信号所在的域可以将语音编码划分为时域编码和频域编码。时域编码是指对语音的时间域信号直接进行编码，例如脉冲编码调制（PCM）、自适应增量调制（ADM）、自适应差分编码（ADPCM）等。频域编码是指对语音的频率域信号进行编码，例如子带编码（SBC）。

按照编码的方法可以分为波形编码、参数编码和混合编码 3 类。波形编码是根据语音信号的波形导出相应的数字编码形式，力求使重建的语音信号波形基本上与输入语音信号波形相同。波形编码具有适应能力强、语音质量好等优点，但编码速率较高，一般在 16～64 kb/s 之间。参数编码又称声源编码或声码器，它是通过对语音信号特征参数的提取和编码，力图使重建的语音信号具有尽可能高的可懂性，重建的音频信号的波形不必与输入的音频信号波形相同。参数编码的优点是编码速率低，而主要问题在于合成音频质量差，自然度较低。通道声码器、线性预测声码器等都属于参数编码器。混合编码是新一代的参数编码算法，这种算法克服了原有波形编码和参数编码的弱点，结合了它们各自的长处，既能保留原语音的自然音色，又具有较高的压缩效率。多脉冲激励线性预测编码（MPLPC）、码本激励线性预测编码（CELP）以及感知编码等都属于混合编码器。

5.2　人的听觉特性

正常人的听觉系统是极其灵敏的，人耳所能感觉到的最低声压接近空气中分子热运动所产生的声压。当声音弱到人的耳朵刚刚可以听见时，我们称此时的声音强度为“听觉阈”。正常人可听声音的频率范围为 16 Hz～16 kHz，年轻人可听到 20 kHz 的声音，而老年人可听到的高频声音减少到 10 kHz 左右。

5.2.1　响度级和响度

声音的响度就是声音的强弱。在物理上，声音响度的客观测量单位是，声压 dyn/cm^2（达因/平方厘米）或声强 W/cm^2（瓦特/平方厘米）。由于人的听觉系统所接受的声强范围很大，因而一般使用对数的形式表示声强（SPL），其单位为分贝（dB），即 $\mathrm{SPL}=10\lg\frac{I}{I_0}$，其中基准声压 $I_0=10^{-16}$ W/cm^2。对应于声波物理模型的声强，在心理上，主观感觉声音的强弱使用响度级“方（Phon）”或者响

度“宋(Sone)”来度量。

确定一个声音的响度级时,需要将它与1kHz的纯音相比较,调节1 kHz纯音的声强,使它听起来与被确定的声音同样响,这时的声压级就规定为该声音的响度级。例如,某噪声的频率为100 Hz,强度为50 dB,其响度与频率为1 kHz,强度为20 dB的声音响度相同,则该噪声的响度级为20方。取响度级为40方(等于响声压级为40 dB的1 kHz纯音)的声音的响度为1宋。方和宋之间的关系为 $S=2^{(P-40)/10}$,或者 $P=33.33\log S+40$。

实验表明,人耳感知的声音响度是频率和声压级的函数,通过比较不同频率和幅度的语音可以得到人耳的等响度曲线,如图5-1所示。图中每条曲线表示等响度时的频率范围。例如,在响度为100方时,对于1 kHz的纯音,相当于10 dB的声压级;而对于100 Hz的纯音,为使它听起来与10方的1 kHz的纯音同样响,则声压级为30 dB,这说明人耳对于不同频率的声音的响应是不平坦的,而且还与声音的响度级有关,声音越大,响应越平坦。图中最下面的一条曲线描述了最小可听声场,也称“绝对听阈”曲线,即在正常情况下,在可听频带范围内,人耳能够感知的最小声压级。

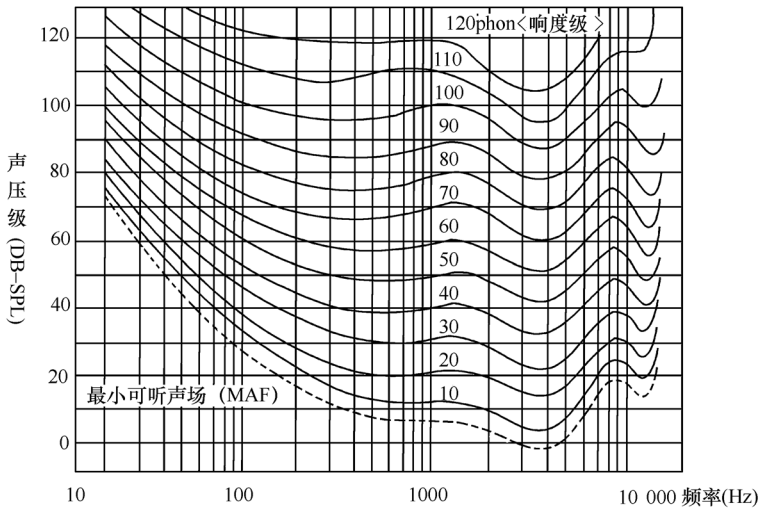


图5-1 等响度曲线与声强的关系

5.2.2 听觉灵敏度

听觉灵敏度曲线表示了在给定频率上,耳朵能够听到声音的最小声压级,如图5-2所示。竖轴表示与这段信号相比其他频率信号能被听到的强度,用分贝表示。从图5-2中可以看出,人耳对不同频率的敏感程度差别很大,其中对2～4 kHz范围的信号最为敏感,在这个频段以外,人耳的听觉灵敏度逐渐降低。一般来说,两个相同能量不同频率的信号,听起来是不一样大的。例如图5-2

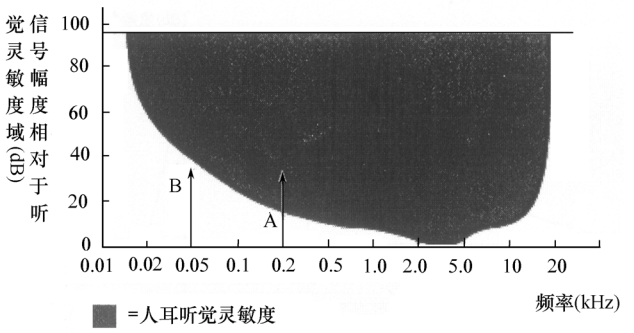


图5-2 听觉灵敏度曲线

中的 A 和 B 强度一样，但只有 A 能被听到，因为 A 在听觉最低灵敏度之上。类似地，人耳能够听到的噪声和失真也随频率而改变。

5.2.3 听觉掩蔽

掩蔽现象是一种常见的心理声学现象，它是由人耳对声音的频率分辨机制决定的。即在一个较强声音的附近，相对较弱的声音将不能被人耳察觉，即被强音所掩蔽。掩蔽效应分为频域掩蔽和时域掩蔽两种。

当音频信号中存在多个信号时，强信号会降低人耳对该信号频率附近其他信号的敏感度，这种现象称为**频域掩蔽**，也称**同时掩蔽**(simultaneous masking)。图 5-3 的曲线表示在一个强信号附近人耳灵敏度的变化。可以看到，由于信号 B 的强度大于信号 A 导致了在信号 B 周围人耳的基本听觉灵敏度曲线被扭曲，这时 A 虽然在单频信号的听觉灵敏度之上也不会被听到。掩蔽效应的大小通常也是随频率变化的，如图 5-4 所示。图中不同曲线表示不同频率信号的掩蔽效应，曲线宽度表示受影响的频率范围，它随频率的增加而增加。

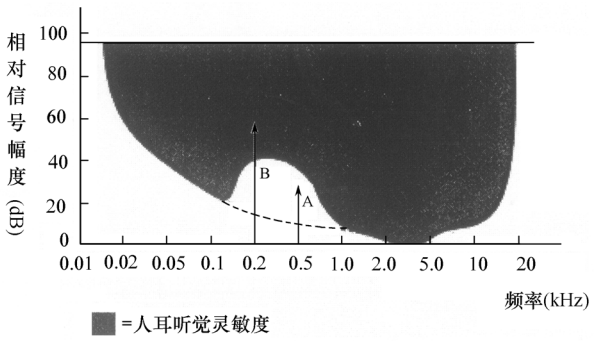


图 5-3 频域掩蔽

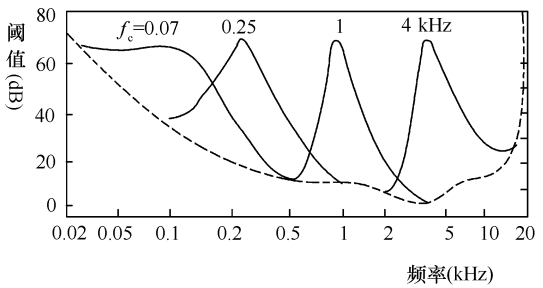


图 5-4 不同频率的掩蔽效应

此外，人耳听到一个强音后，会经过一个短暂的延时才能听到较弱的声音，这称为**时域掩蔽**，也称**异时掩蔽**(non-simultaneous masking)。时域掩蔽分为超前掩蔽(pre－masking)和滞后掩蔽(post－masking)两种。超前掩蔽是指一个信号被前面发出的噪声(或另一个信号)掩蔽；滞后掩蔽则是指一个信号被后面发出的一个噪声(或一个信号)掩蔽。产生时域掩蔽的主要原因是人的大脑需要一定的时间来处理信息。图 5-5 给出了同时掩蔽和时域掩蔽现象，从图中可以看出，同时掩蔽是一种较强的掩蔽效应，异时掩蔽随着时间的推移很快衰减。一般来说，超前掩蔽很短，只有大约 5～20 ms，而滞后掩蔽可以持续 50～200 ms。

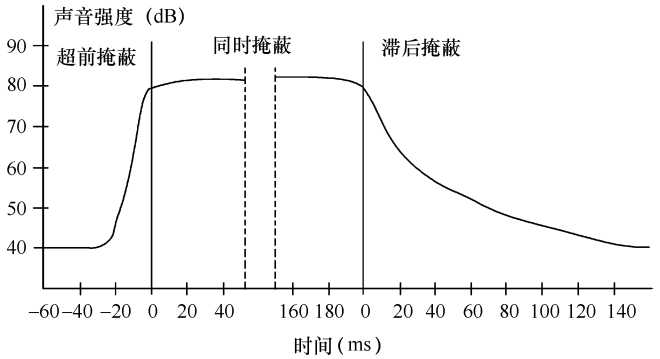


图 5-5 时域掩蔽

频域掩蔽和时域掩蔽作用合并可以形成一个时间－频率曲线，如图 5-6 所示，落入曲线以下的声音都会被掩蔽。人类听觉系统的掩蔽效应需要用一个心理声学模型来描述，依据该模型可以估计出掩蔽阈值，从而能够改变信号，并对其进行合理的编码。

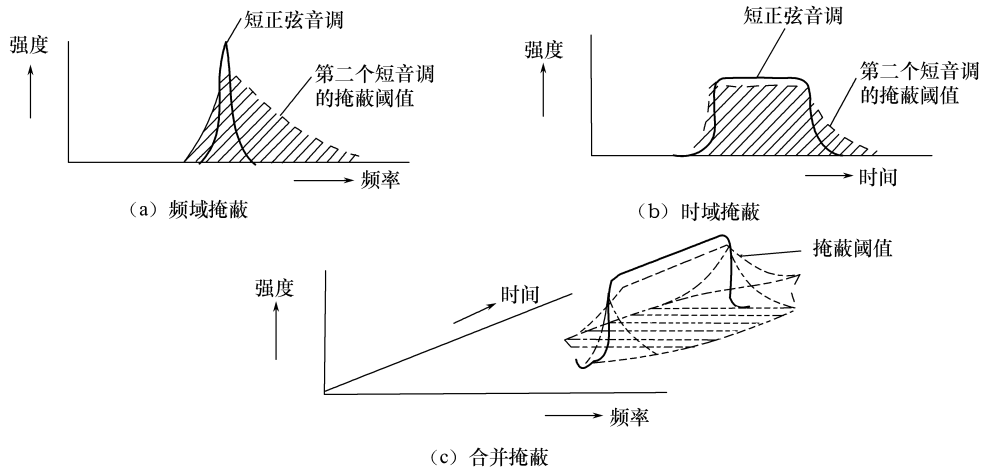


图 5-6　频域和时域掩蔽合并后产生的时间－频率曲线图

5.2.4　临界带宽

为了描述窄带噪声对于纯音调信号的掩蔽效应，人们引入了临界带宽的概念。一个纯音可以被以它为中心频率、并且具有一定频带宽度的连续噪声所掩蔽，如果在这一频带内噪声功率等于该纯音的功率，这时该纯音处于刚好能被听到的临界状态，则称这一带宽为**临界带宽**。实验表明，频率低于 500 Hz 时，临界带宽是约为 100 Hz 的常数。当频率高于 500 Hz 时，临界带宽近似地以 100 Hz 的倍数线性增加。例如，一个 1 kHz(2×500 Hz)的信号，临界带宽大约是 200 Hz(2×100 Hz)；5 kHz 的信号的临界带宽是 1 kHz 等。

临界带宽的单位可以用 Bark(巴克)来表示，1Bark＝一个临界频带宽度。一般我们将 20 Hz～16 kHz之间的频率分为 24 个临界带宽，即 24 个 Bark。关于 24 个临界带宽的编号、中心频率、临界带宽以及频率范围的详细数据见参考文献[1]。

临界带宽的编号 Z(Bark)与频率 f(Hz)之间的关系也可以近似表示为

$$Z \approx 26.81f/(1960+f)-0.53$$

5.3　音频信号的数字化

音频信号可能是自然产生的，也可能是通过电子合成器产生的。电子合成的音频信号是以数字形式产生的，而自然产生的音频信号是模拟信号，需要把它们转换成数字信号。

音频信号的取样频率 f_s 可以依据取样定理来确定；为了保证不产生频谱混叠，还需要在取样之前，利用低通滤波器将被取样信号的上限频率限制在 $f_s/2$ 以下。由于实际的低通滤波器难以具有理想陡峭的截止特性，因此如果实际低通滤波器的截止频率为 f_c，则取样频率应选为 $f_s=(2.2\sim2.5)f_c$。根据各种不同的应用，音频信号的频率上限不同，相应的取样频率可以从 8 kHz 到 192 kHz。实际应用中可以根据所要求的数字电路的速度、存储或传输媒介的容量进行折中。例如，制造厂商选择 44.1 kHz 作为 CD 盘的取样频率，这是考虑到播放时间和媒介造价等几个方面的因素，而 DVD 盘则使用 192 kHz 的取样频率。

在一个二进制数字系统中，字长决定了可以量化的间隔数。n 比特的字长可以提供 2^n 个量化

电平。比特数越多，越接近于模拟值。当量化误差小到某种程度时，可以认为是听不见的。一般认为音频信号采用 16～20 比特的字长就相当令人满意了。当然，这并不排除使用更长的字长和其他的信号处理技术来降低量化误差。例如，DVD 格式采用 24 比特编码，许多音频记录设备中也会采用噪声整形技术来降低带内量化噪声。

根据声音的频带、取样频率和样本精度等，通常把声音的质量分成 5 个等级（见表 5-1），由低到高分别是电话、调幅（AM）广播、调频（FM）广播、激光唱盘（CD）和数字录音带（DAT）的声音。

表 5-1　声音质量和数据率

质量	采样频率(kHz)	精度(bit/样本)	单道声/立体声	数据率(kB/s)（未压缩）	频率范围
电话 *	8	8	单道声	8	200～3 400 Hz
AM	11.025	8	单道声	11.0	20～15 000Hz
FM	22.050	16	立体声	88.2	50～7000Hz
CD	44.1	16	立体声	176.4	20～20 000 Hz
DAT	48	16	立体声	192.0	20～20 000 Hz

* 电话使用 μ 律编码，动态范围为 13 位。

5.4　音频自适应差分脉冲编码调制

语音信号常常采用 3.5.1 节介绍的差分脉冲编码调制（DPCM）的方法进行压缩。在 DPCM 的基础上，加上自适应量化步长调整，使其能够根据信号的幅度来改变差分信号所用的比特数，从而对变化范围小的差分值用较少的比特编码，就可以达到节省带宽或改善信号的质量的目的。这种编码方式就是**自适应差分脉冲编码调制**（Adaptive DPCM，ADPCM）。

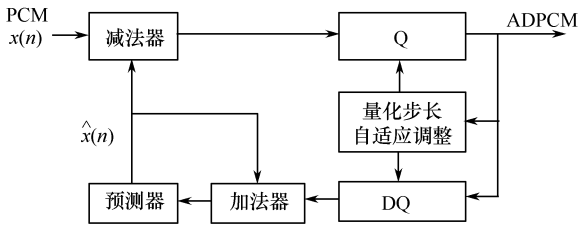


图 5-7　ADPCM 编码器框图

图 5-7 为 ADPCM 编码器框图，它是一种波形编码技术。图中的预测器也是自适应的，预测以帧为单位进行（帧长的典型值为 20ms），根据本帧语音波形的时间相关性确定预测器系数，使预测误差的方差最小。自适应线性预测又可以分为前向预测和后向预测两种，前向预测采用当前帧的取样值计算预测器系数，其预测精度高，代价是引入一帧时间的算法时延。反向预测采用上一帧的样本值计算预测器系数，它虽然没有算法时延，但预测精度较低。

ITU-T 制定的 G.721 标准采用了 32 kb/s 的 ADPCM。在此基础上制定的扩充标准 G.723，可将数据率降低到 40 kb/s 和 24 kb/s。

5.5　音频子带编码

ITU-T 关于 64 kb/s 7 kHz 带宽的高音质音频编码标准 G.722，采用子带编码（见 3.7 节），其编解码框图如图 5-8 所示。输入的 224 kb/s（16 kHz 取样，16 b/样值）的 PCM 信号，经 24 阶的发送正交镜像滤波器组（QMF），将全频带的信号分为两个子带：高子带（4 000～7 000 Hz）分量和低子带（50～4 000 Hz）分量，取样频率为 8 kHz。上子带信号采用 16 kb/s 的 ADPCM 进行编码，下子带采用 48/40/32 kb/s 的 ADPCM 编码，因此称为 **SB-ADPCM** 编码。G.722 可以提供三种不同

的编码速率(对应三种工作模式):64 kb/s、56 kb/s 或 48 kb/s。其中,56 kb/s 或 48 kb/s 的编码速率可以在总体 64 kb/s 的速率中,分别设立一个 8 kb/s 和 16 kb/s 的辅助数据传输通道。解码端分接器将收到的比特流分为高、低两个比特流,分别进行 ADPCM 解码,再经过接收 QMF 合并成 224 kb/s 的 PCM 信号输出。

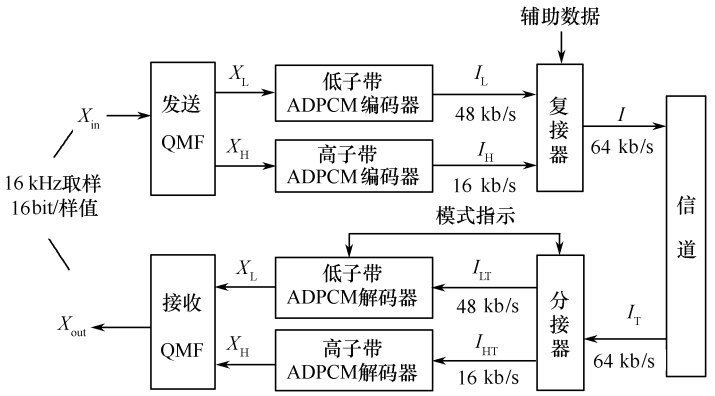


图 5-8　G.722 编解码器方框图

下面我们以工作模式一(64 kb/s)为例,简单介绍两个子带 ADPCM 编码过程。此模式下,低子带的编码速率为 48 kb/s,高子带为 16 kb/s,复接后形成 64 kb/s 的信号。

低子带的 ADPCM 编码器框图如图 5-9 所示。低子带输入信号 X_L 减去其预测值 S_L 得到差值 e_L,用 60 阶的自适应量化器进行量化,产生 48 kb/s 比特流 I_L。在反馈环里,删除输出信号 I_L 的两个最低有效比特,产生一个 4 bit 信号 I_{Lt},I_{Lt}一方面用于调整量化器参数,一方面通过 15 阶逆量化器产生差值信号 d_{Lt},用于自适应预测器对当前值的预测。预测出的信号 S_L 加到量化产生的差值信号上产生低子带输入信号的重建形式 r_{Lt}。重建信号和量化的差值信号都由自适应预测器进行运算,产生输入信号的预测值 S_L,从而完成了反馈环。

图 5-10 是高子带 ADPCM 编码器示意图。其输入信号 X_H 减去预测值 S_H 得到差分信号 e_H,使用 4 阶自适应量化器对其量化产生一个 16 kb/s 的信号 I_H。逆自适应量化器对 I_H 进行处理产生量化的插值信号 d_H。把信号估值 S_H 加到这个量化的差值信号 d_H 上以产生高子带输入信号的重建形式 r_H。重建信号和量化的差值信号由自适应预测器进行运算,产生输入信号的估值 S_H,从而完成了反馈环。

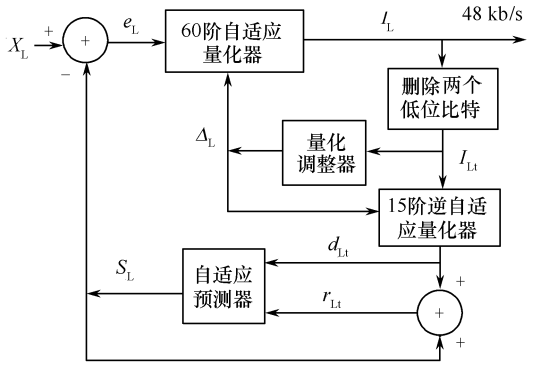


图 5-9　G.722 低子带 ADPCM 编码器方框图

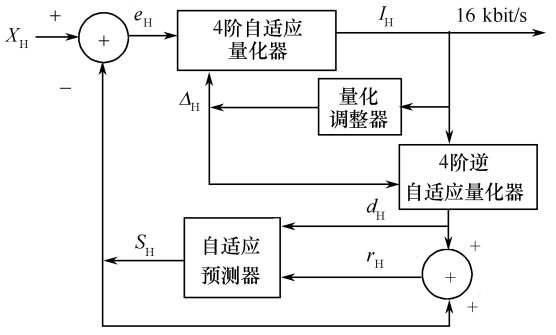


图 5-10　G.722 高子带 ADPCM 编码器方框图

ITU-T 的 G.726 标准同样使用子带编码,输入语音带宽最高为 3.4 kHz,编码后速率为 40、32、24 或 16 kb/s。

5.6 线性预测编码

线性预测编码(Linear Predictive coding,LPC)的原理就是通过分析时间信号波形,提取出其中重要的音频特征,然后将这些特征量化并传送。在接收端用这些特征值重新合成出声音,其质量可以接近于原始信号。LPC 属于参数编码,由于它只传输代表语音信号特征的一些参数,所以可以获得很高的压缩比,4.8 kb/s 就可以实现高质量的语音编码,甚至可以在更低速率(2.4 kb/s 或者 1.2 kb/s)传输较低质量的语音,其缺点则在于人耳可以感觉到再生的声音是合成的。因此 LPC 编解码器主要应用于窄带信道的语音通信和军事领域,因为在这里降低带宽是最重要的。

如何提取听觉特征值是 LPC 算法的核心。根据语音信号的特点,**音调**、**周期**和**响度**这三个特征值可以决定一个语音信号所产生的声音。另外,声音中的浊音(由声带产生,例如与字母 m、v 和 l 相关的声音)和清音(声带展开时产生,例如与字母 f 和 s 相关的声音)也是很重要的参数。一旦从声音波形中获得这些参数,就能够利用合适的声道模型再生原始的语音信号。LPC 编解码器的基本原理如图 5-11 所示。首先将输入信号划分为帧,然后对每帧信号提取听觉参数,将结果编码并传输。编码器的输出是一个帧序列,每帧包含相应的字段,用以表示音调、响度、周期(由取样频率决定)、信号是清音还是浊音的标识以及一组新计算出来的模型系数值。在接收端,由声道模型逐帧再生出语音信号。

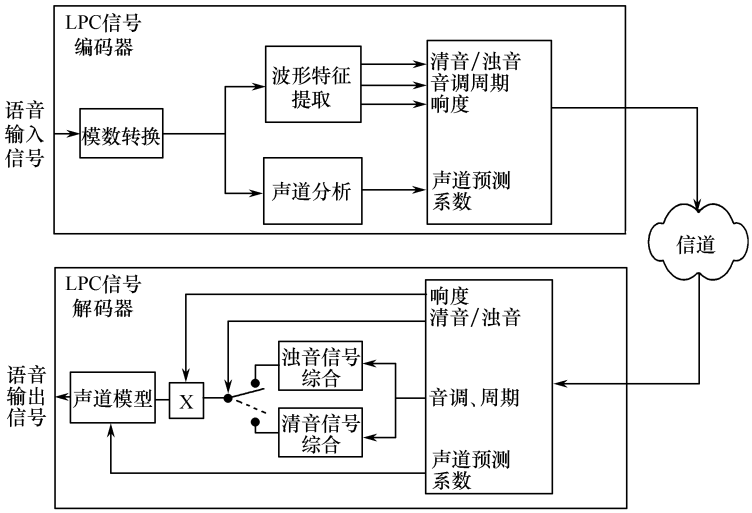


图 5-11 LPC 编解码器原理图

5.7 码激励线性预测编码(CELP)

语音混合编码是在线性预测编码(语音参数编码)的基础上,引入波形编码,使用合成分析法而形成的一种新的编码技术。混合编码克服了参数编码激励形式过于简单的缺点,成功地将波形编码和参数编码两者的优点结合起来,即利用了语音产生模型,通过对模型参数编码,减少了被编码对象的数据量,又使编码过程产生接近原始语音波形的合成语音,以保留说话人的各种自然特征,提高了语音质量。

码激励线性预测编码(Code Excited Linear Prediction,CELP)属于混合编码,它以语音线性预测模型为基础,对残量信号采用矢量量化,利用合成分析法(ABS,Analysis-By-Synthesis)搜索最佳激励码矢量,并采用感知加权均方误差最小判决准则,获得了高质量的合成语音和优良的抗噪声性

能，在 4.8～16 kb/s 的速率上获得了广泛的应用。ITU－T 的 G.728、G.729、G.729（A）和 G.723.1 四个标准都采用这一方法来保证低数据速率下较好的声音质量。

5.7.1 感知加权滤波器

感知加权滤波器(Perceptually Weighted Filter)利用了人耳听觉的掩蔽效应，通过将噪声功率在不同频率上的重新分配来减小主观噪声。在语音频谱中能量较高的频段(共振峰附近)的噪声相对于能量较低的频段的噪声而言，不易被人耳所感知。因此在度量原始输入语音与重建语音(合成语音)之间的误差时，可以充分利用这一现象：在语音能量较高的频段，允许两者的误差大一些；反之则小一些。感知加权滤波器的传递函数为

$$W(z)=\frac{A(z)}{A(z/\gamma)}=\frac{1-\sum_{i=1}^{p}a_i z^{-i}}{1-\sum_{i=1}^{p}a_i \gamma^i z^{-i}} \qquad 0<\gamma<1 \tag{5-1}$$

式中，a_i 为线性预测系数，γ 是感知加权因子。γ 值决定了滤波器 $W(z)$的频率响应，恰当地调整 γ 值可以得到理想的加权效果。最合适的 γ 值一般由主观听觉测试决定，对于 8 kHz 的取样频率，γ 的取值范围通常介于 0.8～0.9。

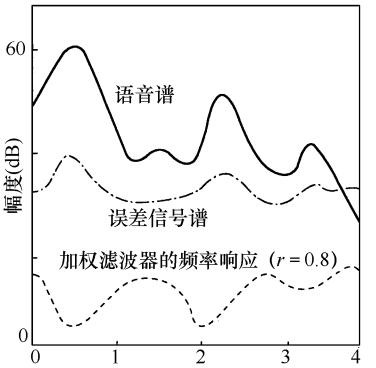


图 5-12 原始语音信号谱、加权后误差信号谱及感知加权 $W(f)$

图 5-12 所示为一段原始输入语音的频谱、经过感知加权滤波器加权后的误差信号频谱，以及感知加权滤波器的频率响应。由图可见，感知加权滤波器的频率响应的峰值、谷值恰好与原始输入语音频率的峰值、谷值相反。这样就使误差度量的优化过程与人耳感觉的掩蔽效应相吻合，产生良好的主观听觉效果。

5.7.2 合成分析法

合成分析法将合成滤波器引入编码器中，使其与感知加权滤波器相结合，在编码器中产生与解码器端完全一致的合成语音，将此合成语音与原始输入语音相比较，调整各相关参数，使得两者之间的感知加权均方误差达到最小，从而提高了重建语音的质量。

合成分析法的原理如图 5-13 所示。与 LPC 编码器相比，在编码器端增加了 LP 合成滤波器和

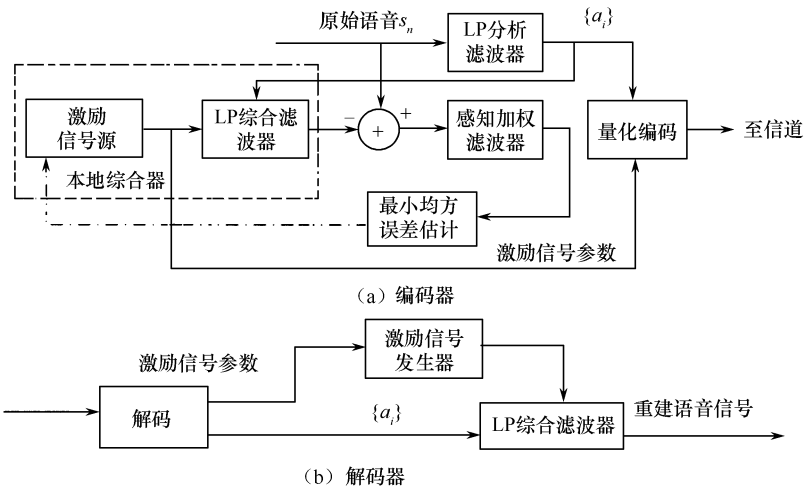


图 5-13 合成分析法原理图

感知加权滤波器。输入的原始语音信号一方面送到 LP 分析滤波器产生预测系数$\{a_i\}$，另一方面与 LP 合成滤波器输出的本地合成语音信号相减，再通过感知加权滤波器，调整激励信号源等相关参数，使原始语音信号与本地合成的语音信号之间误差的感知加权均方值最小，然后将相应的分析参数$\{a_i\}$和激励信号参数进行编码、传送。在解码器端，将信号解码获得$\{a_i\}$及激励信号参数，用这些参数控制调整相应的合成滤波器及激励信号发生器，产生合成语音。

5.7.3 CELP 编解码原理

CELP 是典型的基于合成分析法的编码器，包括基于合成分析法的搜索过程、感知加权、矢量量化和线性预测技术。它从码本中搜索出最佳码矢量，乘以最佳增益，代替线性预测的残差信号作为激励信号源。CELP 采用分帧技术进行编码，帧长一般为 20～30 ms，并将每一语音帧分为 2～5 个子帧，在每个子帧内搜索最佳的码矢量作为激励信号。

CELP 的编码原理如图 5-14 所示，图中虚线框内是 CELP 的激励源和综合滤波器部分。CELP 通常用一个自适应码本中的码字来逼近语音的长时周期性（基音）结构，用一个固定码本中的码字来逼近语音经过短时和长时预测后的差值信号。从两个码本中搜索出来的最佳码字，乘以各自的最佳增益后再相加，其和作为 CELP 的激励信号源。将此激励信号输入到 P 阶合成滤波器，得到合成语音信号$\hat{S}(n)$，$\hat{S}(n)$与原始语音信号$s(n)$之间的误差经过感觉加权滤波器$W(z)$，得到感觉加权误差$e(n)$。通过感知加权最小均方误差准则，选择均方值最小的码字作为最佳的码字。CELP 编码器的计算量主要取决于码本中最佳码字的搜索，而计算复杂度和合成语音质量则与码本的大小有关。

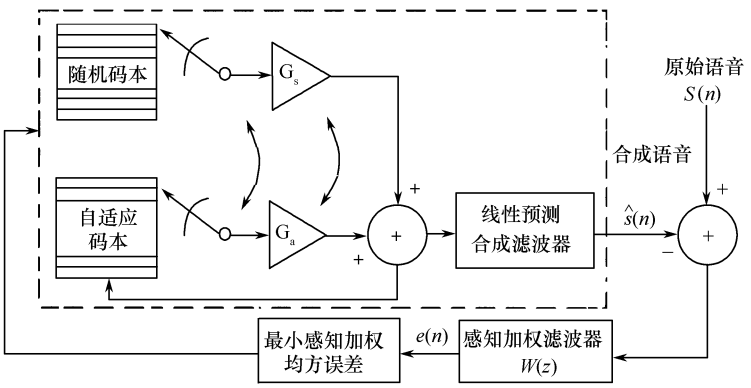


图 5-14　CELP 编码器示意图

CELP 的解码器示意图如图 5-15 所示。解码器一般由两个主要的部分组成：合成滤波器和后置滤波器。合成滤波器生成的合成语音一般要经过后置滤波器滤波，以达到去除噪声的目的。解码的操作也按子帧进行。首先对编码中的索引值执行查表操作，从激励码本中选择对应的码矢量，通过相应的增益控制单元和合成滤波器生成合成语音。由于这样得到的重构语音信号往往仍旧包含可闻噪声，在低码率编码的情况下尤其如此。为了降低噪声，同时又不降低语音质量，一般在解码器中要加入后置滤波器，它能够在听觉不敏感的频域对噪声进行选择性抑制。

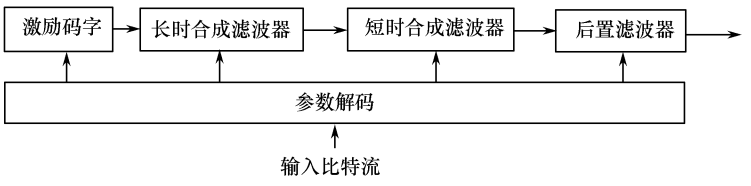


图 5-15　CELP 解码器示意图

5.7.4 G.729 编解码器

G.729 语音编解码器是 ITU－T 制定的 8 kb/s 语音压缩标准，它采用共轭结构代数码激励线性预测(**CS-ACELP**)编码，其原理如图 5-16 所示，其中图 (a)是编码器结构，图(b)为解码器结构。编码器分析出的参数有线性预测系数、基音延迟、基音增益、固定码本激励码字的索引和增益。在解码器中，自适应码本与固定码本的向量分别乘上各自的增益，相加后得到激励信号。将激励信号输入 10 阶的线性预测合成滤波器来重建语音信号。

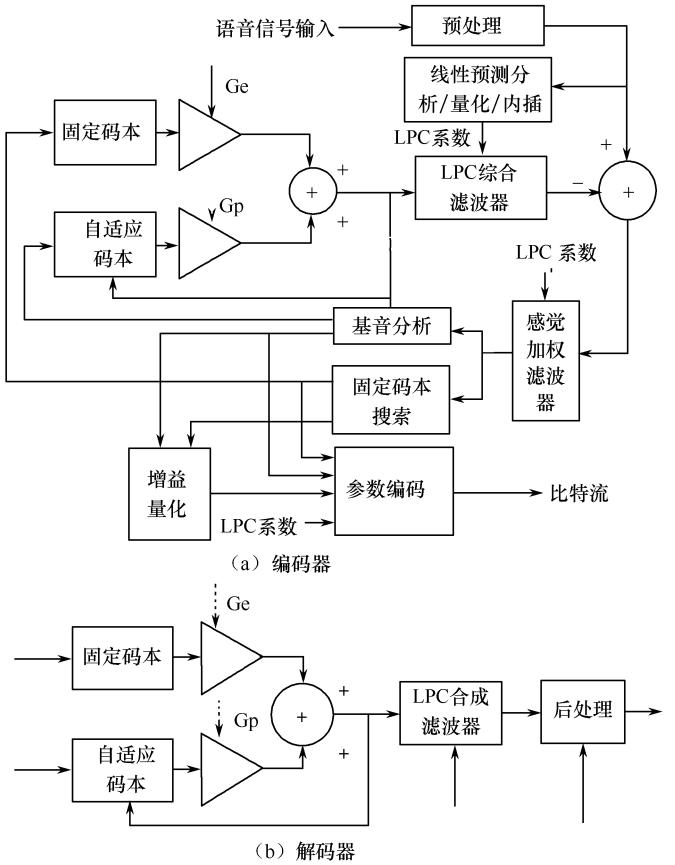


图 5-16　G.729 编解码器原理图

下面我们把 G.729 编解码器的主要部分作一简单介绍，详细情况请参阅相关标准：

(1) 线性预测与量化

编码器首先对输入的 8 kHz 取样的 16 bit PCM 信号进行预处理，然后对每帧(10 ms)语音进行线性预测分析，得到 LPC 系数，并将其转换为 LSP 参数，接着对 LSP 参数进行二级矢量量化。编码器中的预处理模块完成以下两个工作：①缩减信号幅度：将输入信号幅度除以 2，使得在对信号定点运算时，降低溢出机率；②高通滤波器：使用截止频率为 140 Hz 的二阶极点/零点滤波器，滤掉不需要的低频成分。

(2) 基音分析

基音分析是为了获得语音信号的基音时延。为了降低搜索过程的复杂性，整个过程采用开环基音搜索和闭环基音搜索相结合的方法。首先对每帧(10 ms)语音信号作开环基音搜索，得到最佳时延 T 的一个候选 T_{op}，然后根据 T_{op} 在每一个子帧(5 ms)内进行闭环搜索，得到各自的最佳基音

时延参数。

(3) 固定码本结构和搜索

G.729 中规定的固定码本采用了代数结构，因此算法简单，码本不需要存储，其码矢量为 40 维，其中有 4 个非零脉冲，每个脉冲的幅度是+1 或者−1，它们出现的位置如表 5-2 所列。

表 5-2　固定码本结构

脉　冲	符　号	位　置
i_0	S_0：$S_0 \pm 1$	m_0：0,5,10,15,20,25,30,35
i_1	S_1：$S_0 \pm 1$	m_1：1,6,11,16,21,26,31,36
i_2	S_2：$S_0 \pm 1$	m_2：2,7,12,17,22,27,32,37
i_3	S_3：$S_0 \pm 1$	m_3：3,8,13,18,23,28,33,38 4,9,14,19,24,29,34,39

固定码本搜索的目的就是要找到 4 个非零脉冲的位置和幅度，还需要对自适应码本增益和固定码本增益进行量化。除了 LSP 参数每帧更新外，其他编码参数每一子帧更新一次。

固定码本搜索是 G.729 编码器中最耗时的一步，因此有很多相关的研究致力于简化此部分的计算量。在 G.729 中使用了焦点搜索法(Focused Search)，仍然占据很大的计算量，所以一种精简版的 G.729A 是用最深树状搜索法(Depth-First Tree Search)，有效的降低了计算量，相关细节可以参考相关规范。

(4) 解码器

在 G.729 的解码器端，通过对接收到的各种参数标志进行解码得到相应的 10 ms 语音帧编码器的参数，解码器在每一子帧内，对 LSP 系数进行内插，并将其变换为 LP 滤波器的系数，然后依次进行激励生成、语音合成和后置处理工作。

G.729 编解码器的合成语音质量较好，实现复杂度较低，主要用于个人移动通信、低轨道卫星通信和无线通信等领域。

5.8　感知编码

感知编码(Perceptual Coding)原理是利用人耳的听觉特性及心理声学模型，通过剔除人耳不能接收的信息来完成对音频信号的压缩。感知编码器首先对输入信号的频率和幅度进行分析，然后将其与人的听觉感知模型进行比较。编码器利用这个模型来去除音频信号中的不相干和统计冗余部分。感知编码器可以将信道的比特率从 768 kb/s 降至 128 kb/s，将字长从 16 比特/样值减少至平均 2.67 比特/样值，数据量减少了约 83%。尽管这种编码的方法是有损的，但人耳却感觉不到编码信号质量的下降。

感知编码基本框图如图 5-17 所示。其中采用了自适应的量化方法，根据可听度来分配所使用的字长。重要的声音分配多一些比特数来确保可听的完整性，而对于不重要的声音的编码位数就会少一些，不可听的声音根本不进行编码，从而降低了比特率。感知编码中常见的压缩率是 4：1、6：1 或 12：1。

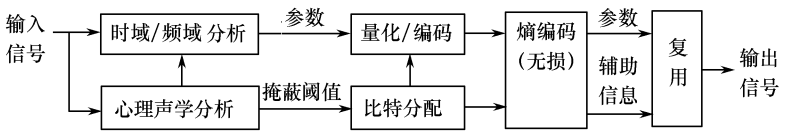


图 5-17　感知编码基本框图

感知编码采用前向自适应分配和后向自适应分配两种比特分配方案。在前向自适应分配方案

中，所有的分配都在编码器中进行，编码信息也包含在比特流中。它的优点是在编码器中采用了心理声学模型，仅仅利用编码数据来完全地重建信号。当改进了编码器中的心理声学模型时，可以利用现有的解码器来重建信号。缺点是需要占用一些比特位来传递分配信息。在后向自适应分配方案中，比特分配信息可以直接从编码的音频信号中推导出来，不需要编码器中详细的分配信息，分配信息也不占用比特位。由于解码器中的比特分配信息是根据有限的信息推导出来的，精度必然会降低。另外解码器相应也比较复杂，而且不能轻易地改变编码器中的心理声学模型。

5.9 MPEG-1 音频编码

5.9.1 概述

MPEG-1 音频系统应用了感知编码和子带编码模型来对声音数据进行压缩。有关音频部分的标准(ISO/IEC11172-3)已经成功的应用在 VCD、CD-ROM、ISDN、视频游戏及数字音频广播中，它支持每声道比特率为 32～224 kb/s 的 32 kHz、44.1 kHz 和 48 kHz 的 PCM 数据，也可以支持带宽为 1.41 Mb/s 下 CD 机的音频编码，以及比特率在 64～448 kb/s 范围内的立体声。

MPEG-1 音频标准的基础是自适应声音掩蔽特性的通用子带综合编码和复用技术(Masking pattern adapted Universal Sub-band Integrated Coding And Multiplexing，**MUSICAM**)与自适应频率感知熵编码(Adaptive Spectral Perceptual Entropy Coding，**ASPEC**)技术。MUSICAM 是比较早且比较成功的感知编码算法。它将输入信号分成 32 个子带，采用基于最小可听阈和掩蔽的感知编码模型达到数据压缩的目的。在取样频率为 48 kHz 时，每个子带的宽度为 750 Hz。根据子带内的 12 个取样值的峰值给每个子带分配 6 比特的标称因子，然后以 0～15 比特的可变字长进行量化。标称因子每隔 24 ms 计算一次，对应 36 个取样值。这里只针对在掩蔽阈值以上的可听信号的子带进行量化，比阈值越高的信号的子带编码的位数也越多，即在给定的比特率下，将比特位分配给最需要的地方，从而得到比较高的信噪比。掩蔽阈值的计算是通过对输入信号进行傅里叶变换得到的。在 128 kb/s 下对 MUSICAM 的测试表明，编码器的保真度与 CD 本身几乎没有差别，至少 2 个编码器级联不会造成音频质量的下降。MUSICAM 编码在复杂度和编码延迟上都比较好，ASPEC 编码器可以保证低数据率下的声音质量。

MPEG-1 音频压缩包括三种模式，称为层次 1、2 和 3。随着层次的增高，复杂度增高，但是各层次之间具有兼容性，即层次 3 的解码器可以对层次 2 或 1 编码的码流进行解码。用户对层次的选择可在复杂性和声音质量之间进行权衡。MPEG-1 第 1 层的复杂度最小，编码器的输出数据率为 384 kb/s，主要用于小型数字盒式磁带；第 2 层采用 MUSICAM 压缩算法，输出数据率为 256～192 kb/s，其应用包括数字声音广播、数字音乐和 VCD 等：第 3 层采用 MUSICAM 和 ASPEC 两种算法的结合，压缩后的比特率为每声道 64 kb/s，其音质仍然非常接近 CD 音乐的水平。

MPEG-1 的声音数据按帧传输，每一帧可以独立解码，如图 5-18 所示。帧的长度由所采用的算法和层决定，第 2、3 层的帧长相同。每帧都包含：①用于同步和记录该帧信息的同步头，长度为 32 比特；②用于检查是否有错误的循环冗余码 CRC，长度为 16 比特；③用于描述比特分配的比特分配域；④比例因子域；⑤子带取样域；⑥有可能添加的附加数据域。帧中最大的部分是子带取样，其个数在不同层次有所不同。例如第 1 层帧有 384 个取样，由 32 个子带分别输出的 12 个样本组成；在第 2、3 层总共有 1152 个取样。

全面的测试表明，与 16 位线性系统相比，不论第 2 层或第 3 层对于 2×128 kb/s 或者192 kb/s 联合立体声音频节目都感觉不到质量的下降；在 384 kb/s 的速率下，第 1 层的效果与 16 比特的 PCM 相同。

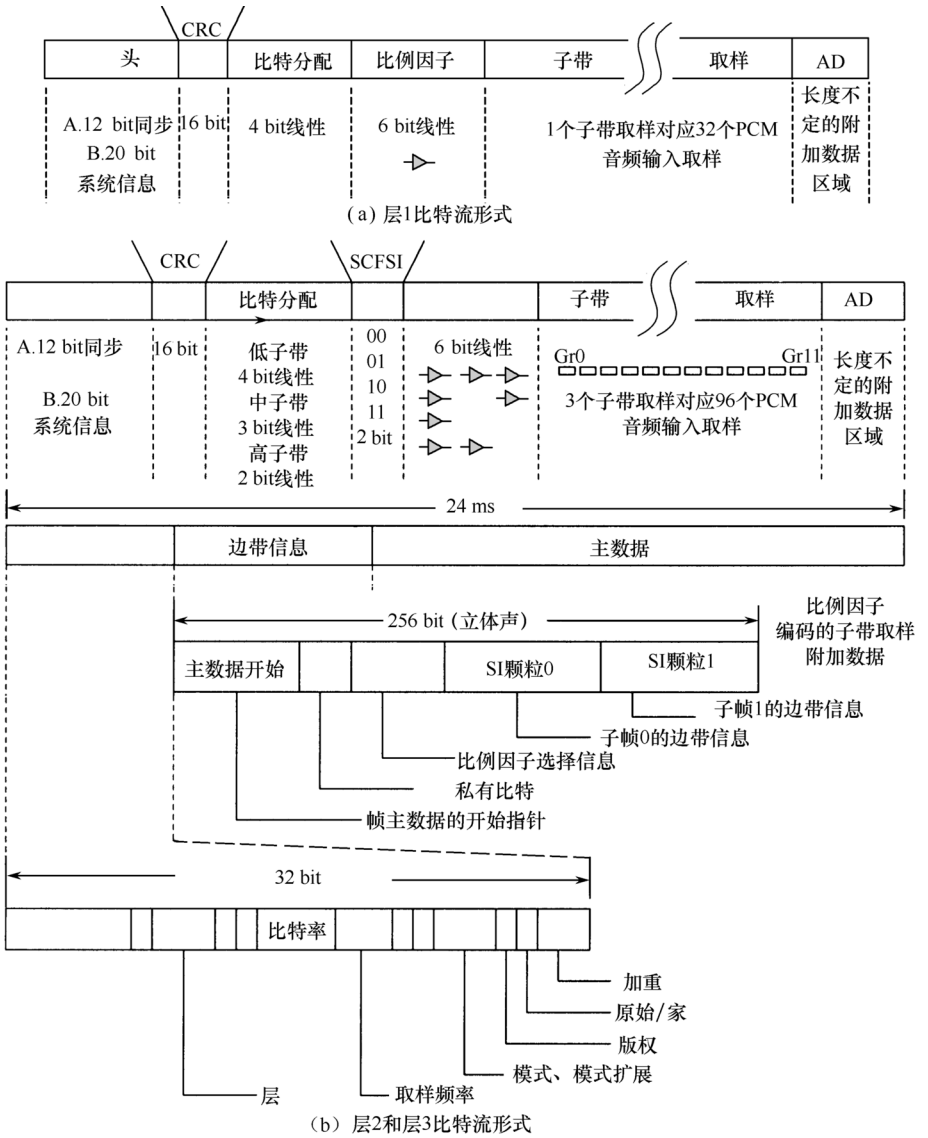


图 5-18　音频第 1、2、3 层的比特流结构

5.9.2　MPEG-1 心理声学模型

MPEG-1 标准建议了两种决定最小掩蔽阈值的心理声学模型。规定只在编码器中使用该模型，而且对于一些简单的编码器可以不使用心理声学模型。在比特分配中利用最大信号电平和掩蔽阈值之间的差异来设定量化级。通常模型 1 用在层 1 和层 2 中，模型 2 用在层 3 中。在这两种情况下，都有一个算法来输出各个子带或者子带组的**信号掩蔽比**(Signal-to-Mask Ratio，SMR)。下面以模型 1 中的算法为例进行说明：

(1) 进行时域到频率的映射。采用 512 或 1024 点的 FFT 并加汉明窗来减少边界效应，将时域数据转换到频域，以精确计算掩蔽值；

(2) 确定最大声压级。在每个子带内根据比例因子和频谱数据进行计算。在确定掩蔽阈值时采用取最大的方法；

(3) 确定安静阈值,形成最低掩蔽边界;

(4) 识别音调和非音调成分。由于信号中的音调和非音调成分的掩蔽阈值不同,首先要识别音调和非音调成分,然后分别来进行处理;

(5) 掩模换算,得到相关的掩模;

(6) 计算掩蔽阈值。每个子带噪声的掩蔽阈值由信号的掩蔽曲线决定。当子带相对于临界带宽比较宽时,选择最小阈值;当其比较窄时,将覆盖子带的阈值进行平均;

(7) 计算全局掩蔽阈值。全局阈值通过对相应的各子带掩蔽阈值和安静阈值求和得到;

(8) 确定最小掩蔽阈值。基于全局掩蔽阈值来确定每个子带的最小掩蔽阈值;

(9) 计算信号掩蔽比。最大信号电平和最小掩蔽阈值之间的差异决定了每个子带的信号掩蔽比值,这个值将用于比特分配。

5.9.3 编码层次

MPEG-1 音频标准中 3 层的基本模型是相同的。其中层 1 是最基础的,层 2 和层 3 都在层 1 的基础上有所提高。每个后继的层都有更高的压缩比,但需要更复杂的编解码器。下面分别进行讨论。

1. 第 1 层

层 1 采用的是简化的 MUSICAM 标准。心理声学模型仅使用了频域掩蔽特性。图 5-19 是单声道第 1 层编解码器的方框图。它可以在较高的数据率下得到高的保真度。

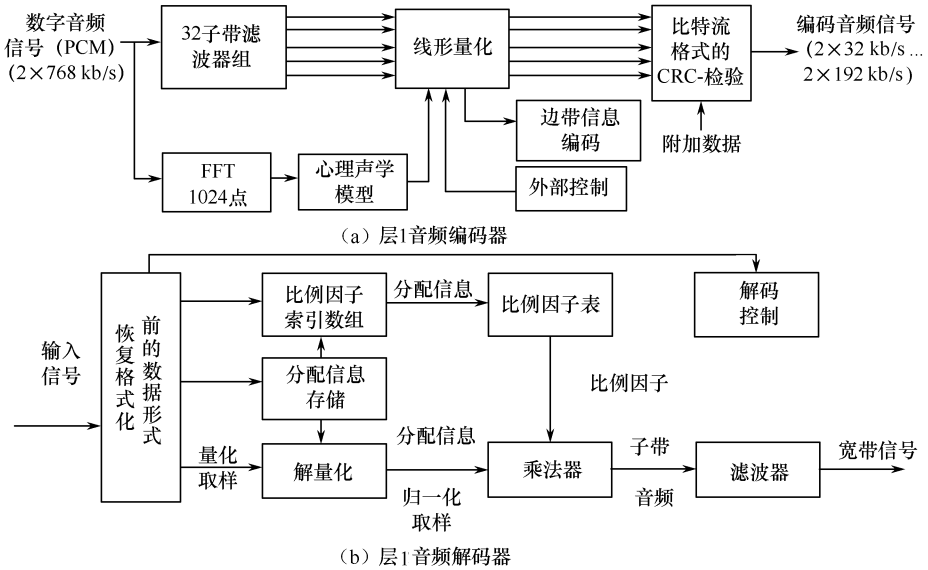


图 5-19 MPEG-1 层 Ⅰ 音频编码器和解码器

编码器的输入可以是模拟的,也可以是以 32 kHz、44.1 kHz、48 kHz 为取样频率的 PCM 数字信号。在这三种取样频率下,子带带宽分别是 500、689 和 750 Hz,帧周期分别是 12、8.7 和 8 ms。

在编码的时候,首先使用多相位滤波器组将音频信号分成 32 个等宽的子带,并进行取样;然后将 32 个子带中的每 12 个子带分成一组形成 1 帧,对每一组的 12 个取样进行比特分配,不可听的子带不进行分配。MPEG-1 的三层中都使用这种 32 子带的多相位滤波器组。

采用浮点数对样值编码,尾数决定分辨率,整数决定动态范围。当某个子带不需要编码时,

则传输一个空的分配信息，在这种情况下，整数和尾数都不传输。对于每一个经过比特分配的子带，根据它的最大样值计算比例因子来对 1 帧中的 12 个子带进行归一化处理。因为音频信号与取样频率相比是慢变化的，掩蔽阈值和比例因子对于每一组的 12 个取样只计算一次。

心理声学模型通过对比例因子信息以及对 512 点 FFT 的频谱分析，将得到的数据与最小阈值曲线进行比较，通过比特分配器对归一化的样值进行量化，达到数据压缩的目的。对于信号掩蔽比 SMR 比值比较大的那些子带多分配一些字长，较小的子带则少分配一些字长。即 SMR 值决定了对子带取样量化的最小信噪比。量化是循环进行的，必要时通过在码字中增加附加比特的方法将 S/N 提升到最小值以上。

层 1 的解码过程如图 5-19(b)所示，一帧一帧地对比特分配信息及比例因子进行解码。通过子带解码器，利用比特分配信息和比例因子将数据重新排列成线性 PCM。收到的比例因子放置在一个 32 行 2 列的数组里，每个数据占用 6 bit。每一列代表一个输出的信道，每一行代表一个子带。将子带样值与比例因子相乘以恢复原始的量化值，空的子带自动赋零。合成重建滤波器将这 32 个子带合并成一个宽带音频信号。由于在解码器中没有使用心理声学模型、比特分配和其他的运算，它的造价很低。尤其重要的是，对于改进的编码器而言，解码器是透明的。

2. 第 2 层

层 2 对层 1 做了一些直观的改进，相当于三个层 1 的帧，每帧有 1152 个样本。在心理声学模型同时使用频域掩蔽和时间掩蔽特性，并且在低、中和高频段对比特分配作了一些限制，对比特分配、比例因子和量化样本值的编码也更紧凑，这样就可以用更多的比特来表示声音数据，音质也比层 1 更高。

图 5-20 给出了层 2 的详细编码过程。滤波器产生 32 个等宽的子带，帧的尺寸为 3×12×32，对应每个信道有 1152 个宽带取样。每个子带的数据按照三组 12 个样值来编码(层 1 使用 1 组)。FFT 增加到 1024 点以便更好地判断音调和非音调成分。

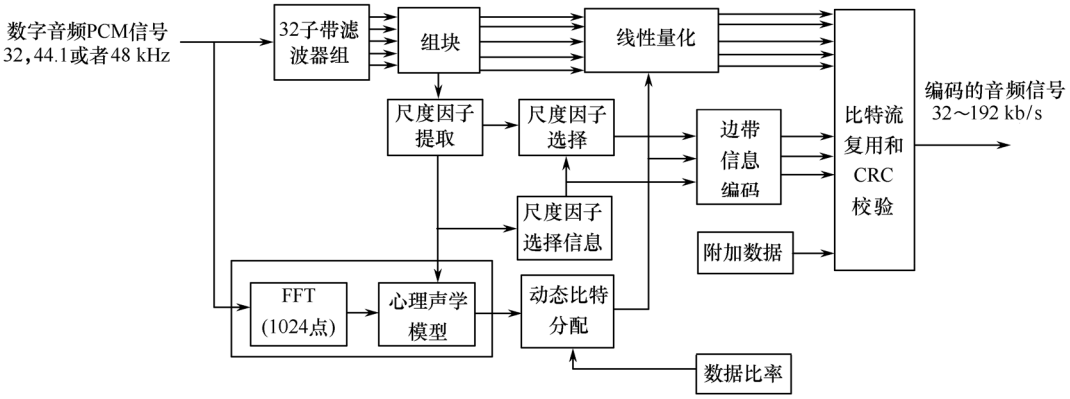


图 5-20　MPEG-1 层 2 音频编码器

对每组的 12 个子带样值进行比特分配，每个子带最多有三个比例因子，每一个对应一组 12 个子带样值。为了减小比例因子的比特率，当组间差别很小或出现时间掩蔽时则共用比例因子。量化范围从 3～65 535 比特，由子带决定可以得到的量化级。例如低的子带最多采用 15 比特，中等的子带则为 7 比特，比较高的子带限制在 3 比特。在每一个频段，给突出的信号分配较长的码字。为了保证高效率，这里将三个连续的样值组成一个区组进行量化。

和层 1 一样，层 2 的解码过程也比较简单。解码器将数据帧解包，然后将数据送入重建滤波器。层 2 编码可以使用立体声编码，通过动态范围控制来适应不同的听众，而且使用了固定长度的

数据字段。最小编码和解码延迟分别为 30 ms 和 10 ms。表 5-3 中对层 1 和层 2 进行了比较。

表 5-3　MPEG-1 音频第 1、2 层参数比较

参　　数	MPEG-第 1 层	MPEG-第 2 层
帧长(取样)	384	1152
子带	32	32
子带取样	12	36
FFT	512	1 024
比特分配	4 per	2 至 4(由子带决定)
尺度因子选择信息	无	2/子带
尺度因子	6/子带	6 至 18 /子带
取样组	无	3 /子带

3. 第 3 层

MPEG-1 第 3 层合并了 MUSICAM 和 ASPEC 算法，相比第 1、2 层都要复杂，在低数据率时能够得到好的保真度。第 3 层的输出就是通常所说的 **MP3**。层 3 使用了比较好的临界频带滤波器，把声音频带分成非等带宽的子带，心理声学模型除了使用频域掩蔽特性和时间掩蔽特性之外，还考虑了立体声数据的冗余，并且使用了霍夫曼编码器。虽然层 3 所用的滤波器组与层 1 和层 2 所用的滤波器组的结构相同，但是层 3 还使用了修正的离散余弦变换 MDCT(见 3.7.3 节)，对层 1 和层 2 的滤波器组的不足作了一些补偿。MDCT 把子带的输出在频域里进一步细分以达到更高的频域分辨率，同时也部分消除了多相滤波器组引入的混叠效应。图 5-21 是单信道时 MPEG-1 第 3 层的编码器和解码器的原理图。

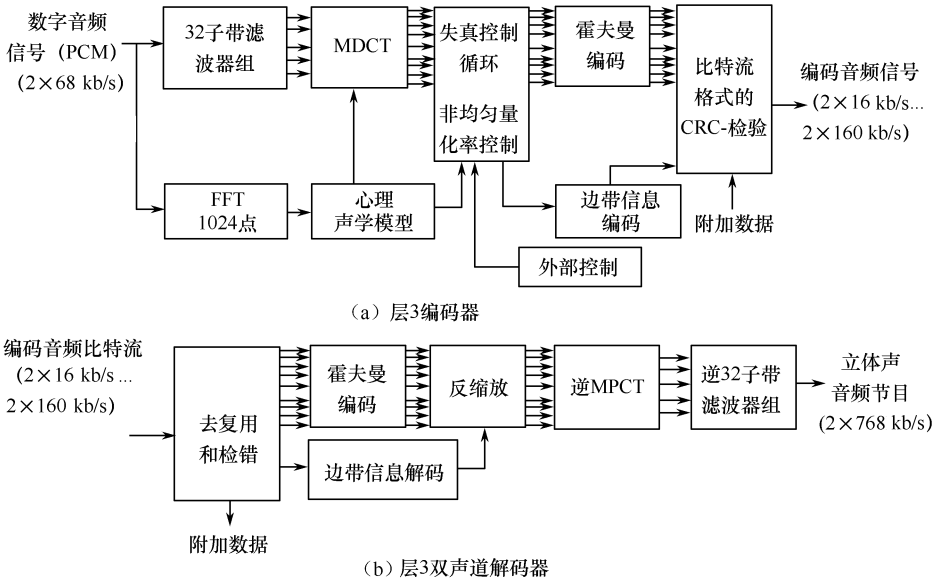


图 5-21　MPEG-1 第 3 层的编码器和解码器

层 3 指定了两种 MDCT 的块长：长块的块长为 18 个样本，短块的块长为 6 个样本，相邻变换窗口之间有 50%的重叠。长块对于平稳的声音信号可以得到更高的频域分辨率，而短块对跳变的声音信号可以得到更高的时域分辨率。在短块模式下，三个短块代替一个长块，而短块的大小恰好是一个长块的 1/3，所以 MDCT 的样本数不受块长的影响。对于给定的一帧声音信号，MDCT 可以全部使用长块或全部使用短块，也可以长短块混合使用。因为低频区的频域分辨率对音质有重

大影响，所以在混合块模式下，MDCT 对最低频的 2 个子带使用长块，而对其余的 30 个子带使用短块。这样既能保证低频区的频域分辨率，又不会牺牲高频区的时域分辨率。长块和短块之间的切换有一个过程，一般用一个带特殊长转短或短转长数据窗口的长块来完成这个长短块之间的切换。

在分配控制算法中采用动态量化，用噪声分配迭代循环方法来计算每个子带的最佳量化噪声。采用分析合成方法计算量化的频谱，以满足掩蔽阈值模型要求的噪声。频谱的量化过程可以根据所观察的比特率的限制来调整，采用非均匀量化法。然后根据音频信号的统计特性采用霍夫曼编码或游程编码达到数据压缩的目的。

在 MPEG-1 第 3 层中，帧与帧之间的数据速率可以改变，这可以应用于变速率录音。先根据心理声学模型计算出需要多少比特位，然后设定帧比特率。变比特率对于多媒体应用中的点播传输很有效。当然变比特流不能通过恒定数据率的系统进行实时传输。当要求恒定速率时，第 3 层允许通过一个可选择的比特转换进行更准确的编码，使平均传输数据率比峰值数据率小。虽然每帧的比特数都变化，但是在长程平均上基本保持恒定。

为了利用立体声信道间的冗余，第 3 层支持 **MS 编码**和立体声编码。MS 编码使用矩阵运算，因此也把 MS 编码称为**矩阵立体声编码**(Matrixed Stereo Coding)。MS 编码不传送左、右声道信号，左、右信号在量化之前分别混合成两者之和与两者之差，以避免立体声中断屏蔽。“和”信号用于中央 M(**m**iddle)声道，“差”信号用于边 S(**s**ide)声道，因此 M/S 编码也叫做“和-差编码(sum-difference coding)”。另外，当左、右信道之间的相关性很高时，可以减小差信号来节约比特位。在编码中，为了节约比特位，只传输每一个比例因子的方向信息。编码可能有的缺陷是造成立体声场的变化，尤其对于瞬态信号。第 3 层允许在 4 种立体声编码方法中进行选择，即左、右信道独立的立体声编码、整个频谱都采用 MS 编码的 MS 模式、低频部分编码为 L/R、高频部分采用强度编码的强度编码模式，以及低频采用 MS 编码、高频部分采用强度编码的 MS 与强度编码混合模式。

第 3 层的解码器采用霍夫曼解码解出比特分配信息。在逆变换中利用频谱系数，在合成滤波器中将 32 个子带合并成一个音频信号。

5.10 MPEG-2 音频编码

MPEG-2 定义了两种声音数据压缩格式，一种称为 MPEG-2 Audio，或者称为 MPEG-2 多通道声音，因为它与 MPEG-1 音频是兼容的，所以又称为 **MPEG-2 BC** (Backward Compatible)；另一种称为 **MPEG-2 AAC** (Advanced Audio Coding)，因为它与 MPEG-1 声音格式不兼容，因此也称为非后向兼容 MPEG-2 NBC (Non-Backward-Compatible)。MPEG 组织于 1994 年制定的 ISO/IEC 13818-3 和 1997 年制定的 ISO/IEC 13818-7 分别给出上述两种标准。

5.10.1 MPEG-2 BC

MPEG-2 BC 使用与 MPEG-1 音频标准相同种类的编解码器，层 1、层 2 和层 3 的结构也相同。在很多情况下，为应用 MPEG-1 设计的算法也适用于 MPEG-2。由于后向兼容性，MPEG-2 的解码器可以接收 MPEG-1 的比特流，MPEG-1 的解码器可以从 MPEG-2 比特流中得到立体声。

与 MPEG-1 标准相比，MPEG-2 音频标准做了如下扩充：①增加了 16 kHz、22.05 kHz 和 24 kHz 取样频率；②扩展了编码器的输出速率范围，由 32～384 kb/s 扩展到 8～640 kb/s；③增加了声道数，支持 5.1 声道和 7.1 声道的环绕声。此外 MPEG-2 还支持线性 PCM 和杜比 AC-3 编码(见 5.11 节)。表 5-4 总结了 MPEG-2 音频标准和 MPEG-1 音频标准的差别。

表 5-4　MPEG-1 和 MPEG-2 的声音数据规格

名称	MPEG-2 Audio	MPEG-1 Audio
取样频率	16/22.05/24/ 32/44.1/48 kHz	32/44.1/48 kHz
样本精度 （每个样本的比特数）	16	16
最大数据传输率	8～640 kb/s	32～448 kb/s
最大声道数	5.1/7.1	2

MPEG-2 的 1、2、3 层提供了三个附加的低频取样频率 16 kHz、22.05 kHz 和 24 kHz，这些附加频率不向后兼容 MPEG-1，这部分标准就是 **MPEG-2 LSF**。层 3 对于这些低比特率表现出了较好的性能。一般只需对 MPEG-1 的比特流和比特分配表作很小的改变，就可以适应这种 LSF 格式。MPEG-2 质量上的改进是通过多相位滤波器组在低频和中频区域改进频率分辨率，从而高效地利用掩蔽效应完成的。

在多声道的 MPEG-2 形式中采用基本的 5 声道方法，有时也称为 3/2 ＋1 立体声（三个前端、两个环绕扬声器声道和一个亚低音扬声器）。亚低音扬声器的低频音效加强声道（Low Frequency Effects，LFE）是可选的，其提供的信号范围最高可至 120 Hz。采用分层的形式使 3/2 多声道可以向下混合为 3/1、3/0、2/2、2/1、2/0 和 1/0 的较低电平音频格式。多声道 MPEG-2 形式采用编码器矩阵，允许用一个双声道解码器对兼容的双声道信号进行解码，这个信号是多声道比特流的一个子集。多声道的 MPEG-2 像其他 MPEG-2 声道一样兼容 MPEG-1 的左、右声道的立体声信息，如图 5-22 所示。附加的多声道数据放置在附加的辅助数据区中。标准的双声道解码器将辅助信息忽略，只重建主要的声道。解码器中的解码矩阵电路在有些情况下会产生附加效应，即声道中的声音被抵消掉，却保留下量化噪声。

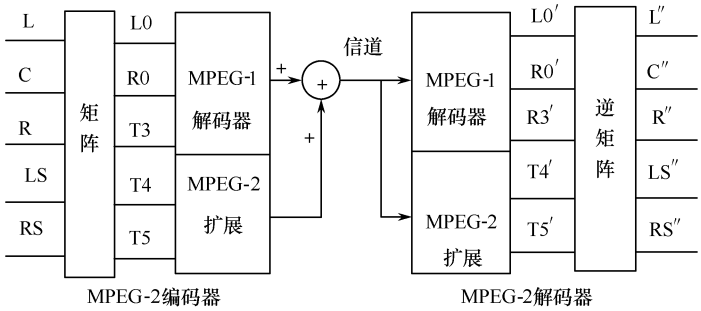


图 5-22　MPEG-2 中的音频编解码器中采用的 5.1 环绕形式

5.10.2　MPEG-2 AAC

MPEG-2 AAC 是 MPEG-2 标准中的一种非常灵活的声音感知编码标准。就像所有感知编码一样，MPEG-2 AAC 主要使用听觉系统的掩蔽特性来减少声音的数据量，并且通过把量化噪声分散到各个子带中，用全局信号把噪声掩蔽掉。

MPEG-2 AAC 编码除去不向后兼容 MPEG-1 之外，其他的性能均比 MPEG-2 BC 优越。它支持 32 kHz、44.1 kHz、48 kHz 取样频率，也支持其他8～96 kHz 的取样频率，并可支持 48 个主声道、16 个低频音效加强通道 LFE、16 个配音声道（overdub channel）或者叫做多语言声道（multilingual channel）和 16 个数据流。MPEG-2 AAC 在压缩比为 11∶1，即每个声道的数据率为（44.1×16 ）/11＝64 kb/s，而 5 个声道的总数据率为 320 kb/s 的情况下，很难区分还原后的声音与原始声音之间的差别。为了改进误差性能，系统的设计能够在噪声存在时保持比特流同步，从而能很好地进行噪声抵消。

图 5-23(a)和(b)分别给出了 MPEG-2 AAC 编码器和解码器的方框图。为了保证输出音频质量的灵活性,MPEG-2 AAC 定义了三种配置:基本配置、低复杂性配置和可变采样率配置。

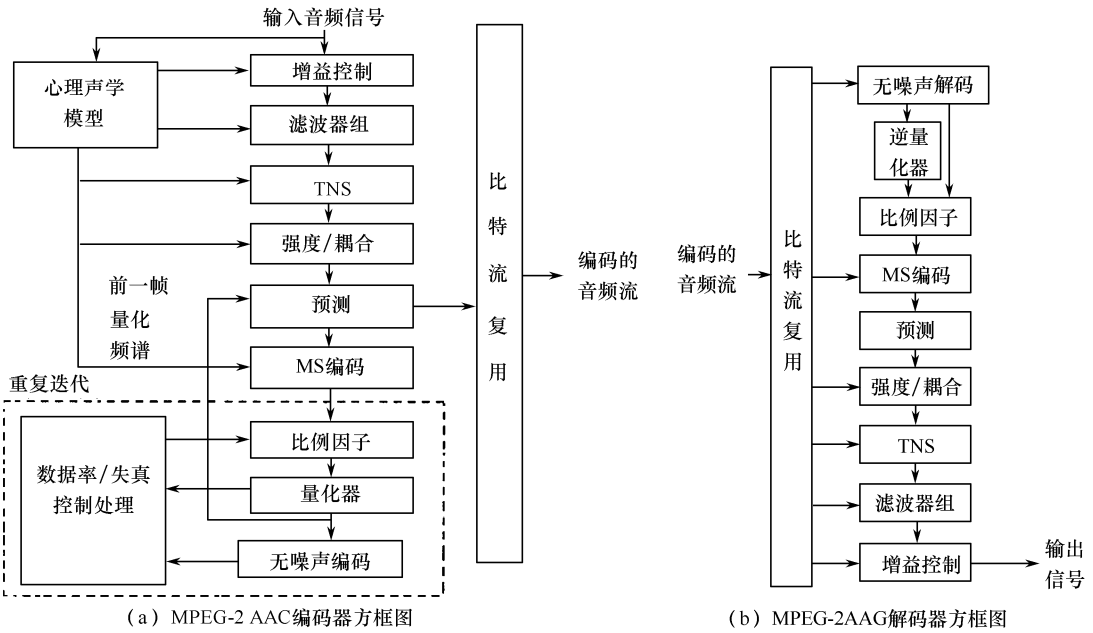


图 5-23　MPEG-2 AAC 编解码器方框图

(1) 基本配置(Main Profile)

在这种配置中,除了"增益控制"模块之外,MPEG-2 AAC 系统使用了图 5-23 中所示的所有模块,在三种配置中提供了最好的声音质量,而且 MPEG-2 AAC 的解码器可以对低复杂性配置编码的声音数据进行解码,但对计算机的存储器和处理能力的要求方面,基本配置比低复杂性配置的要求高。

(2) 低复杂性配置(Low Complexity Profile)

在这种配置中,不使用预测模块和预处理模块,瞬时噪声成形(Temporal Noise Shaping,TNS)滤波器的级数也有限,这就使声音质量比基本配置的声音质量低,但对计算机的存储器和处理能力的要求可明显减少。

(3) 可变采样率配置(Scalable Sampling Rate Profile)

在这种配置中,使用增益控制对信号作预处理,不使用预测模块,TNS 滤波器的级数和带宽也都有限制,因此它比基本配置和低复杂性配置更简单,可用来提供可变采样频率信号。

下面对 MPEG-2 AAC 的基本模块作一些简单的介绍。

(1) 增益控制模块

在可变取样率配置中采用该模块,它由多相正交滤波器 PQF(polyphase quadrature filter)、增益检测器和增益修正器组成,其作用是把输入信号分离到 4 个相等带宽的频带中。

(2) 滤波器组

滤波器组将输入信号从时域变换到频域,它采用了修正离散余弦变换 MDCT 和时域混迭消除 TDAC(见 3.7.3 节)技术。两个相邻的窗互相重叠 50%。当感知的重要成分间隔小于 140 Hz 时使用正弦窗,当各成分间距大于 220 Hz 时,使用 Kaiser-Bessel 窗。窗函数的切换是无缝的。对于取样频率可伸缩的情况,在解码器中可以忽略上部 PQF 频带来得到低取样率信号。例如,18 kHz、12 kHz、6 kHz 的带宽可以通过忽略 1、2、3 频带来得到。

(3) 瞬时噪声成形 TNS

在感知编码中,瞬时噪声成形 TNS 模块是用来控制量化噪声的瞬时形状的一种方法,解决掩蔽阈值和量化噪声的错误匹配问题。这种技术的基本想法是,在时域中的音调声信号在频域中有一个瞬时尖峰,TNS 使用这种双重性来扩展已知的预测编码技术,把量化噪声置于实际的信号之下以避免错误匹配。

(4) 立体声编码技术

MPEG-2 AAC 中使用了两种立体声编码技术,即强度编码和 MS 编码。这两种方法可以根据信号频谱选择使用,也可以混合使用。

(5) 预测和量化器

MPEG-2 AAC 中用一个 2 阶后向自适应预测器来消除平稳信号的冗余,它采用非均匀的量化。

(6) 无噪声编码

无噪声编码实际上就是霍夫曼编码,它对量化的频谱系数、比例因子和方向信息进行编码。

5.11 杜比(AC-3)编码

杜比 **AC-3** 是一种灵活的音频数据压缩技术,它具有将多种声轨格式编码为一种低码率比特流的能力。支持 8 种不同的声道配置方式,从传统的单声道、立体声,到拥有 6 个分离声道的环绕声格式(左声道、中置声道、右声道、左环绕声道、右环绕声道及低音效果声道)。AC-3 的比特流可以支持 48 kHz、44.1 kHz、或 32 kHz 三种取样频率,所支持的码率从 32 kb/s 到 640 kb/s 不等。

杜比 AC-3 编码系统属于感知编码器,采用 MDCT 的自适应变换编码算法,利用临界频带内一个声音对另一个声音信号的掩蔽效应最明显,将整个音频频带分割成若干个较窄的频段,划分频带的滤波器组要有足够锐利的频率响应,以保证临界频带外的噪声衰减足够大,使时域和频域内的噪声限定在掩蔽门限以下。由于人类的听觉对不同频率的声音具有不同的灵敏度,因此各频段的宽度并不完全一样,每一个频段所占有的数据量不是平均分配的。编码器通过人耳的听觉掩蔽特性,根据信号的动态特性来决定在某一时刻的数据应当如何分配给各个频段。对于频谱密集、音量大的声音元素应该获得较多的数据占有量,而那些由于掩蔽效应而听不到的声音则少占用或不占用数据量。

AC-3 编码器原理如图 5-24 所示。在编码器中首先采用 TDAC 滤波器(见 3.7.3 节)把时域内的 PCM 取样数据变换到频域,每个变换系数以二进制指数形式表示,即由一个指数和一个尾数构成。指数部分反映信号的频谱包络,经编码后构成了整个信号大致的频谱。频谱包络在比特分配过程中用来决定对每个尾数编码所需要的比特数。最后由 6 个音频块的频谱包络、粗量化的尾数及相应的参数组成 AC-3 数据帧格式,打包并进行传输。

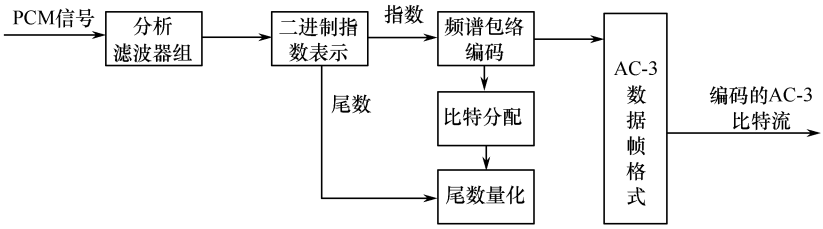


图 5-24 AC-3 编码器原理图

由时域变换到频域的块长度的选择是变换编码的基础,在 AC-3 中定义了两种由时域变换到频域的块长度,一种是 512 个样值的长块,一种是 256 个样值的短块。对于稳态信号,其频率随时

间变换缓慢，为提高编码效率，要求滤波器组有好的频率分辨力，即要求一个长块；而对于快速变化的信号，则要求好的时间分辨力，即要求一个短块。在编码器中，输入信号在经过 3 Hz 高通滤波器去除直流成分后，再经过一个 8 kHz 的高通滤波器取出高频成分，用其能量与预先设定的阈值相比较，通过检测信号的瞬变情况来选择变换的块长度。

在 AC-3 编码器的比特分配中，采用了前向和后向自适应比特分配法则。前向自适应方法是编码器计算比特分配，并把比特分配信息明确地编入数据比特流中，其特点是在编码过程中使用听觉模型，因此修改模型对接收端的解码过程没有影响；其缺点是由于要传送比特分配信息而占用了一部分有效比特，降低了编码效率。后向自适应方法不需要得到编码器明确的比特分配信息，而是从比特流中产生比特分配信息。其优点是不占用有效比特，因此有更高的传输效率；缺点在于由于要从接收的数据中计算比特分配，如果计算太复杂会使解码器的成本升高；此外，解码器的算法也会随着编码器听觉模型的改变而改变。由于 AC-3 中采用了前/后向混合自适应比特分配方法以及公共比特池等技术，在提高码率和降低成本之间进行了折中，因而可使有限的码率在各声道之间、不同的频率分量之间获得合理的分配。

AC-3 的比特流是由帧构成的，如图 5-25 所示。在恒定的时间间隔内，其所有编码的声道所包含的信息就能体现在 1 536（6 个编码的块，每块的 256 个取样）个 PCM 采样值的信息。每一个 AC-3 的帧都具有固定的尺寸，只由采样频率及编码数据率决定。同时，每个帧都是独立的实体，且不与前一个帧分享数据，除了在 MDCT 所固有的去交迭变换。在每个AC-3帧的开头是 SI 域（Sync Information）及 BSI 域（比特流信息）。SI 域及 BSI 域描述了比特流的结构，包括采样频率、数据码率、编码声道的数目及其他一些系统描述的元素。每个帧都包含两个 CRC 用于错误检测，一个在帧的末尾，一个在 SI 的头中，由解码器选择 CRC 来进行校验。在每个帧的结尾处有一个可选的辅助数据域。在这个区域内允许系统设计者在 AC-3 比特流中嵌入可在整个系统内传递的、自有的控制字及状态字信息。

同步	CRC #1	SI	BSI	音频块 0	音频块 1	音频块 2	音频块 3	音频块 4	音频块 5	辅助数据	CRC #2

图 5-25　AC-3 帧结构

每个帧有 6 个声音块，每个块表示为每个编码声道包含 256 个 PCM 取样，如图 5-26 所示。声音块中的内容包括块转换标志、耦合坐标、指数、比特分配参数、尾数等。

块交换标志（Block Switch Flags）	抖动标志（Dither Flags）	动态范围控制（Dynamic Range Control）	耦合策略（Coupling Strategy）	耦合坐标（Coupling Coordinates）	指数策略（Exponent Strategy）	指数（Exponents）	比特分配参数（Bit Allocation Parameters）	尾数（Mantissa）

图 5-26　AC-3 声音块结构

AC-3 解码器如图 5-27 所示。首先解码器必须与编码数据流同步，然后从经过数据纠错校验的比特流中分离出控制数据、系统配置参数、编码后的频谱包络及量化后的尾数等内容，根据声音的频谱包络产生比特分配信息，对尾数部分进行反量化，恢复变换系数的指数和尾数，再经过合成滤波器组，把数据由频域变换到时域，最后输出重建的 PCM 样值信号。

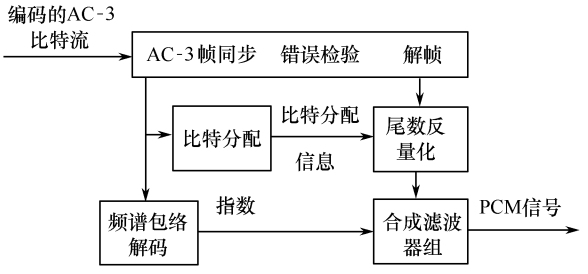


图 5-27　AC-3 解码器原理图

AC-3 的编/解码器被设计成一个完整的音频子系统的解决方案，它拥有普通的低码率编/解码所没有的许多特性。这些特性包括适用于消费类音频回放系统的动态范围压缩特性、对话归一(Dialog Normalization)以及缩混特性(Downmixing)。其中缩混特性可以将多声道音频进行转换为特定数目的声道输出。动态范围控制的控制字是嵌入在 AC-3 比特流内，在解码器中使用，可以使同一个比特流源在不同模式下进行还音。杜比 AC-3 技术中充分利用了人耳的听觉模型，针对不同性质的信号，采取了相应有效的算法，达到了在保证较高音质的前提下实现较高压缩率的目的，是一种非常高效而又经济的数字音频压缩系统。AC-3 是美国数字电视系统的强制标准，是欧洲数字电视系统的推荐标准。

5.12 音频压缩编码的国际标准

语音编码技术在近十几年得到了突破性的发展，出现了许多实用的高质量的语音编码算法。针对不同的应用，国际电联 ITU、MPEG 标准化组织以及一些地区标准协会已经制定了一系列的语音编码标准。

1. ITU－T 语音编码标准

ITU－T 制定的语音编码的标准为 G 系列标准，具体如表 5-5 所示。其中 G.726 是 G.721 与 G.723 的合成，G.726 被推出后，G.721 和 G.723 就被删除了。

表 5-5 波形编码国际标准

标准	制定时间	语音质量	编码速率 (kb/s)	编码算法	复杂度		延迟时间 (ms)
					MIPS	RAM (字节)	
G.711	1972	长途电话	64	μ/A 律 PCM	＜＜1	1	0.125
G.726 (G.721 G.723)	1988 (1984 1986)	长途电话	40/32/24/16	ADPCM	1.25	＜50	0.125
G.727	1990	长途电话	40/32/24/16	ADPCM	1.25	＜50	0.125
G.722	1988	长途电话	64/56/48	SBC	10	1 K	1.5
G.728	1994	长途电话	16	CELP	30	2 K	0.625
G.723.1	1995	＜长途电话	5.3,6.3	多脉冲,CELP	20	2 K	30
G.729	1996	长途电话	8	CELP	25	3 K	10

2. 地区性语音编码标准

随着移动通信技术的迅速发展和普遍应用，也出现了一些针对应用的地区性标准。其中有代表性的主要包括欧洲的数字蜂窝电话语音编码器、北美的数字蜂窝语音编码器和日本的数字蜂窝电话编码器。如表 5-6 所示。

(1) GSM 全速率 RPE-LTP 编码器

这个编码器是欧洲的特别移动小组 GSM 于 1987 年为全欧洲数字蜂窝电话而制定的。RPE-LTP(Regular-Pulse Excited Linear Predictive Coding)表示有长期预测器的规则脉冲激励线性预测器。GSM 全速率通道支持 22.8 kb/s，其中附加的 9.8 kb/s 用于信道编码，来保护编码器在无线信道中所产生的比特错误。除了数字蜂窝电话的应用外，这个编码器的低复杂度可以用于情报传送。

(2) GSM 半速率语音编码器

这个编码器是为了将 GSM 蜂窝系统的容量加倍而提出的，采用了摩托罗拉公司设计的 5.6 kb/s 矢量和激励线性预测编码器 VSELP(Vector Sum Exhibition)，通道中的很大一部分用于误码保护。

(3) 自适应多速率 AMR 编码器

AMR(Adaptive Multi-Rate,自适应多速率)编码是欧洲电信标准化协会(ETSI)为 GSM 系统所制定的语音压缩编码标准,已被 3GPP 采纳作为第三代移动通信 WCDMA 系统的语音压缩编码标准。AMR 的基本概念是以更加智能的方式解决信源和信道编码的速率分配问题,使得无线资源的配置和利用更加灵活和高效。

AMR 中采用了代数码激励线性预测技术(ACELP),编码器输入为 8 kHz,16 比特量化的线性 PCM 编码,编码操作以 20 ms 语音为 1 帧。发送端编码器提取 ACELP 模型参数进行传输,接收端译码器再根据这些参数构成的激励信号合成出重建的语音信号。

AMR 标准支持八种不同码速的语音流:12.2 kb/s,10.2 kb/s,7.95 kb/s,7.4 kb/s,6.7 kb/s,5.9 kb/s,5.15 kb/s 和 4.75 kb/s,具体的编码原理参见参考文献[2,3]。这八种编码模式可使 AMR 标准应用于不同架构或版本的移动网络。与 GSM 语音编码采用固定的编码速率相比,在实际应用中,AMR 编码标准可以根据无线信道和传输状况来自适应地选择一种最佳编码模式,以保证语音通信的正常进行。

(4) IS-54 7.95 kb/s VSELP

这个编码器是北美电信工业联盟 TIA 为北美的时分多址(TDMA)数字蜂窝电话制定的。这个编码器在失真性能上的质量指标与 RPE-LTP 相比还是有一些差距。

(5) IS-96 8.5 kb/s 变速率码激励线性预测 QCELP

变速率码激励线性预测 QCELP 编码器是 TIA 为北美的码分多址数字蜂窝电话制定的标准。它是 IS-96 的一部分,用在 IS-96 所规定的系统中。QCELP 是一个可变速率的编码器,它使用数字语音内插技术,以获得数据速率的降低。在通话时间内,它以 8.5 kb/s 的速率运行;当通道内不存在语音的时候,它的速率会降低到 0.8 kb/s。

(6) 日本的数字蜂窝语音编码器 JDC-VSELP

这个标准是日本无线系统研究和发展中心(RCR)为日本的 TDMA 数字蜂窝系统制定的标准。它与 IS-54 VSELP 非常相似。JDC-VSELP 比 RPE-LTP 和 IS-54 两个编码器在性能上的要求要低一些。

(7) JDC 半速率 3.45 kb/s 音调同步 CELP(PSI-CELP)语音编码器

这个标准是为了让日本的 TDMA 个人数字蜂窝系统(PDC)容量加倍而制定的。采用了音调同步的 CELP 革新方案,使用自适应固定码书,以改善低速率语音编码器的语音质量。

表 5-6　地区性标准

标准	制定机构	制定时间	语音质量	编码速率(kb/s)	编码算法	延迟时间(ms)
GSM 全速率	ETSI	1987	接近长途电话	13	RPE－LTP	20
GSM 半速率	ETSI	1994	接近长途电话	5.6	VSELP	24.375
AMR	ETSI	1999	通信	12.2/10.2/7.95/7.4/6.7 /5.9/5.15/4.75	ACELP	25
IS－54	ITA	1989	接近长途电话	7.95	VSELP	20
IS－96	ITA	1993	IS－54	8.5/4/4/0.8	QCELP	20
JDC 全速率	RCR(日)	1990	IS－54	6.7	VSELP	25
JDC 半速率	RCR(日)	1993	同全速率	3.45	PSI-CELP	50

3. MPEG 制定的音频标准

前面我们已经详细介绍了 MPEG 组织制定的 MPEG-1 和 MPEG-2 音频标准。MPEG-1 的音频信号标准包括三种压缩模式,即层次 1、2 和 3。随着层次的增高,复杂度增高,各层次之间具有兼容性。MPEG-2 种包含两种声音数据压缩格式 MPEG-2 BC 和 MPEG-2 AAC。

4. AC-3 音频压缩标准

AC-3 标准通常应用于数字电视广播和 HDTV 系统的音频数据压缩，它是一种灵活的音频数据压缩技术，具有将多种声轨格式编码为一种低码率比特流的能力。

习 题 五

1. 假设我们使用下面的预测器：

$$\hat{f}_n = \text{trunc}\left[\frac{1}{2}(\tilde{f}_{n-1} + \tilde{f}_{n-2})\right] e_n = f_n - \hat{f}_n$$

并且使用量化方程 $\tilde{e}_n = Q[e_n] = 16 * \text{trunc}[(255 + e_n)/16] - 256 + 8$，如果输入信号值如下：20，38，56，74，92，110，128，146，164，182，200，218，236，254；DPCM 编码器的输出如下（不含熵编码）：20，44，56，74，89，105，121，153，161，181，195，212，243，251。

(1)编写程序实现该算法，并验证结果；

(2)假设在无损编码中，由于我们使用预测器时发生了错误，使用原始信号值而不是量化信号值，所以在解码端，我们获得了如下的重构信号：20，44，56，74，89，105，121，137，153，169，185，201，217，233。可以看出误差在逐渐加剧，编写程序验证结果。

2. 编程实现一个 CELP 编码器，用你自己录制的声音数据测试该编码器。

3. 如果在 MPEG 音频第一层的取样频率是 32 kb/s，那么每一个 32 子带中的子带带宽是多大？

4. 自己录制一段声音数据取样，取出其中连续的 36 个样本信号，分别实现 MDCT 和 DCT，比较两者得到的频率。当声音频率较低时，哪种方法能更好地把能量集中到前面几个系数？

参 考 文 献

[1] [美]ken c. pohlmann 著. 数字音频原理与应用. 苏菲译. 北京：电子工业出版社，2002

[2] 高成伟. 移动多媒体技术——标准、理论与实践. 北京：清华大学出版社，2006

[3] 王炳锡. 变速率语音编码. 西安：西安电子科技大学出版社，2004

第 6 章　多媒体同步

同步是在各类通信系统中经常遇到的一个概念，它往往与统一的时间基准（或者说时钟）相关联。例如收、发端的同步表示收、发端时钟是同频率和同相位的；网同步表示全网有统一的时钟等。而本章所讨论的是**媒体同步**，它虽然与时钟同步有密切的关系，但是所包含的概念更为广泛。媒体同步是由多媒体数据所具有的独特特征而引发出的问题，换句说话，只有在多媒体系统中才有多媒体同步的问题。也正因为如此，本章对多媒体同步的讨论需要从介绍多媒体数据的特点开始。

6.1　多媒体数据

6.1.1　连续媒体数据与静态媒体数据

已在 1.2 节中讲过，多媒体数据是由在内容上相互关联的文本、图形、图像、动画、语音和活动图像等媒体数据构成的一种复合信息实体。多媒体数据的形成过程，就是这些不同类型数据在计算机的控制之下合成的过程。在这一过程中，每一种媒体数据都是以数字化的方式被表示、存储、传输和处理的。其中，有着严格时间关系的音频、视频等类型的数据称为**实时媒体**数据或**连续媒体**（Continuous Medium）数据，其他类型的数据被称做**非实时媒体**数据、**离散媒体**（Discrete Medium）数据或者**静态媒体**数据。一般地讲，在涉及多媒体数据时，意味着这种复合数据体中至少包含一种非实时数据和一种实时数据。

虽然不同媒体类型的数据都可以表示为数字信号，但其特点各不相同。按数据对时间的敏感性和数据生成方式的差别，可以将不同媒体类型的数据划分为表 6-1 所示的几类。

表 6-1　媒体数据的成分

时间敏感性 / 生成方式	连续媒体（敏感）	静态媒体（不敏感）
获取（源自现实世界）	声音、视频信号	静止图像
合成（由计算机完成）	动画	文本、图形

声音、视频数据和静止图像数据通常是由某种采集设备（如麦克风、摄像机、扫描仪等）直接获取，经 A/D 转换后进入计算机系统的，这称为获取数据；由计算机生成的动画、文本、图形等数据则称为**合成（Synthetic）数据**。不过，随着语音合成技术、光字符识别技术 OCR（Optical Character Recognition）等新技术的应用，这种划分的界线变得越来越模糊。

连续数据可以看成是由**逻辑数据单元** LDU（Logical Data Unit）构成的时间序列，或称为**流**（Stream）。LDU 的划分由具体的应用、编码方式、数据的存储方式和传输方式等因素决定。例如，对于符合 H.262 标准的视频码流，一个 LDU 可以是一个宏块、一个条、一帧图像，或者是构成一个场景的几帧图像（如图 6-1 所示）等。连续数据的各个 LDU 之间存在着固定的时间关系，例如以一

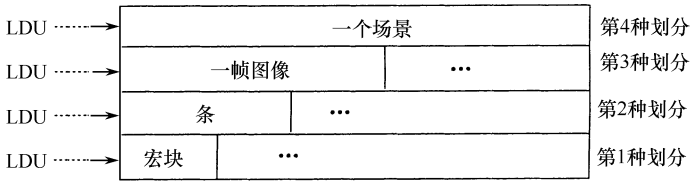


图 6-1　H.262 码流中 LDU 的不同划分

帧图像为一个 LDU，则相继的 LDU 之间的时间间隔为 40 ms(见图 6-2)。这种时间关系是在数据获取时确定的，而且要在经过存储、处理、传输之后在播放过程中保持不变，否则就会损伤媒体显示时的质量，例如产生图像的停顿、跳动，或声音的间断等。在静态数据内部则不存在这种时间关系。

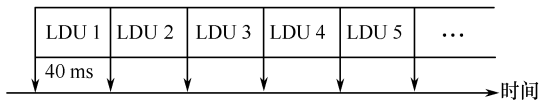


图 6-2　连续媒体 LDU 之间的相对时间关系

6.1.2　多媒体数据内部的约束关系

多媒体数据所包含的各种媒体对象并不是相互独立的，它们之间存在着多种相互制约的关系(或称**同步关系**)。反之，毫无联系的不同媒体的数据所构成的集合不能称为多媒体数据。多媒体数据内部所固有的约束关系可以概括为基于内容的约束关系、空域约束关系和时域约束关系[1]。

1. 基于内容的约束关系

基于内容的约束关系是指，在用不同的媒体对象代表同一内容的不同表现形式时，内容与表现形式之间所具有的约束关系。这种约束关系在数值分析中应用得比较多，例如对原始数据进行分析的结果可以用报表、图形、或者动画的形式反映在最终提交给用户的多媒体文档中。由于人们对于不同类型的媒体有着不同的感受，如报表给人以精确详尽的感觉，图形显得直观，而动画则能让人更好地了解数据的演变过程，因此采用多种表现形式能够使用户对于原始数据有一个全面的认识。

为了支持这种约束关系，多媒体系统需要解决的主要问题是，在多媒体数据的更新过程中确保不同媒体对象所含信息的一致性，即在数据更新后，保证代表不同表现形式的各媒体对象都与更新后的数据对应。解决这一问题的一种办法是，定义原始数据和不同类型媒体之间的转换原则，并由系统而不是由用户来完成对多媒体文档内容的调整。

2. 空域约束关系

空域约束关系又称为**布局(Layout)关系**，它用来定义在多媒体数据显示过程中的某一时刻，不同媒体对象在输出设备(如显示器、纸张等)上的空间位置关系。这种约束关系是排版、电子出版物与著作等系统中要解决的首要问题。由这些系统生成的多媒体文档被称为**结构化文档**。

办公室文档结构 ODA(Office Document Architecture)是一种定义结构化文档的国际标准，由 ISO 制定(ISO 8613 系列)，后为 ITU 所支持，并更名为开放性文档结构(T.410 协议系列)。ODA 标准主要针对办公环境下常见的文档类型(如信件、报告、备忘录等)，以及由文字处理程序生成的文档(可包含文本、图形、图像)而制定。早期的 ODA 标准不支持声音、活动图像等连续媒体；经扩展后的标准 HyperODA 文档可以支持声音、活动图像、超级链以及对各数据体之间时域关系的定义[2]。

ODA 定义了逻辑文档结构和布局文档结构，并采用树状模型对这两种结构进行层次化描述。文档内容(即各媒体对象)被存放在叶子中，叶子的属性表明了数据的媒体类型。文档的逻辑结构表示内容的组织方式，如章节、标题、注解等；布局结构则描述了各数据体之间的空域关系。媒体对象和基本布局对象之间存在着确定的映射关系，而基本布局对象和输出设备的某一矩形区域相对应，其位置可以根据输出设备上的某一固定点或与其他基本布局对象的相对关系来标注。如图 6-3所示，多个基本布局对象又可构成复合布局对象，从而形成表示媒体对象间空域关系的树状结构。

3. 时域约束关系

时域约束关系(或称时域特征)反映媒体对象在时间上的相对依赖关系,它主要表现在如下两个方面:

(1) 连续媒体对象的各个 LDU 之间的相对时间关系;

(2) 各个媒体对象(包括连续媒体对象以及静态媒体对象)之间的相对时间关系。

连续媒体对象内部 LDU 之间的时间约束关系已在图 6-2 中作了说明。图 6-4 给出了媒体对象之间的相对时间关系的例子。图中表示声音 1 和电视图像同时播放,继而播放三幅静止图像(P_1、P_2、P_3),然后播放一段动画,动画期间插入声音 2。

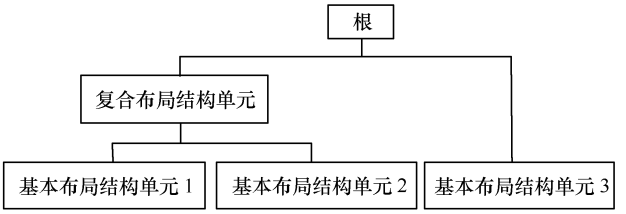


图 6-3 ODA 的布局文档结构

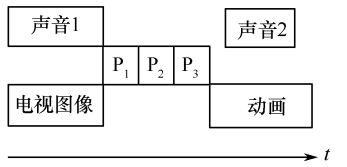


图 6-4 不同媒体对象之间的时间约束关系

媒体对象之间的时域约束关系按照确立这种关系的时间来区分,可以分为**实时**(Live)**同步**和**综合**(Synthetic)**同步**两种。实时同步是指在信息获取过程中建立的同步关系。例如,人物口形动作和声音之间的配合,通常称为**口形**(或唇)**同步**(Lip-Sync);又如,当处于不同地点的多个与会者在各自的计算机上观看同一幅图表,其中一人用箭头指着图表作解说时,出现在其他人的屏幕上的箭头必须和解说一致,这称为**指针同步**(Pointer-Sync)。口形同步与指针同步都属于实时同步。综合同步是指在分别获取不同的信息之后,再人为地指定的同步关系。在播放时,系统将根据指定的同步关系显示有关的信息。在图 6-4 所示的例子中,录像片断、3 幅静止图像和动画之间的串联顺序就属于综合同步关系。综合同步可以事先定义,也可以在系统的运行过程中定义。例如在一个导游系统中,根据用户即时输入的要求,系统自动地产生对某条旅游路线的解说,配合介绍该条路线的录像也同时播放。解说与录像之间的时间约束关系就是在运行过程中指定并执行的。

在上述 3 种约束关系中,时域特征是最重要的一种。当时域特征遭到破坏时,用户就可能遗漏或者误解多媒体数据所要表达的信息内容。例如在观看体育比赛的现场直播时,电视画面的暂时中断或不连贯,会妨碍观众对比赛过程的准确了解,而这种画面的中断或不连贯就是时域特征遭到破坏的具体表现。由此可以理解,时域特征是多媒体数据语义的一个重要组成部分。时域特征被破坏,也就破坏了多媒体数据语义的完整性。在本章后面的叙述中,将只讨论有关时域约束关系方面的问题。

6.1.3 多媒体数据的构成

根据上面的讨论,多媒体数据的构成可以用图 6-5 来表示。其中主体部分是不同媒体(如文字、图形、图像、声音和活动图像)的数据,这些数据包含了所要表达的信息内容,称为构成多媒体数据的**成分数据**。除了成分数据之外,它们之间的约束关系(同步关系)也是构成多媒体数据的不可缺少的组成部分。这些约束关系称为**同步规范**(Synchronization Specifications)。在存储和传输成分数据时,必须同时存储和传输它们之间的同步关系。在对成分数据作处理时,

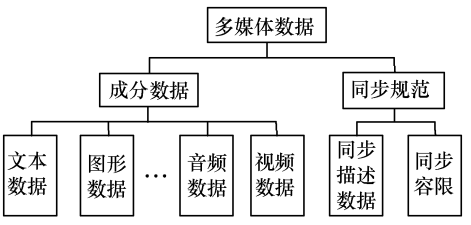


图 6-5 多媒体数据的构成

必须维持它们之间的同步关系。当只考虑时域同步关系时，时域同步规范由**同步描述数据**和**同步容限**两部分组成。同步描述数据表示媒体内部和媒体之间的时间约束关系，同步容限则表示这些约束关系所允许的偏差范围。

上述结构反映了多媒体数据与传统的计算机数据的本质区别，并由此产生了多媒体系统中的同步问题。多媒体同步所研究的主要问题是：

(1) 如何表示(描述)多媒体数据的时域特征；

(2) 在处理多媒体数据的过程中(如采集、传输、播放等)，如何维持时域特征。完成第二项工作的机制称为**同步机制**。

6.2 多媒体数据时域特征的表示

6.2.1 时域场景和时域定义方案

对多媒体数据的时域特征进行抽象、描述以及给出必要的同步容限，是在表示时域特征的过程中所要完成的任务。这里，抽象是一个忽略与时域特征不相干的细节(如数据量、压缩及编码方式等)、将多媒体数据概括为一个时域场景的过程。一个时域场景由若干时域事件构成，每一个时域事件都是与多媒体数据在时域中发生的某个行为(如开始播放、暂停、恢复以及终止播放等)相对应的。时域事件可以认为是瞬时完成的(例如在第 6 秒开始播放一段电视图像等)，也可以认为是持续一段时间的(例如播放过程持续 6 分钟等)。如果一个时域事件在场景中的位置可以完全地确定，称该事件为**确定性时域事件**，否则就是**非确定性时域事件**。例如，暂停播放、恢复播放等事件在时域场景中的位置不能事先确定，只有在播放多媒体对象的过程中，才能够根据用户交互的实际情况确定下来。凡是包含有非确定性时域事件的场景为非确定性时域场景，反之则为确定性时域场景。例如对图 6-6 所示的两个时域场景来说，图(a)场景中不含有任何非确定性时域事件，因而是确定性时域场景。在图(b)场景中，由于 p 事件和 r 事件的位置有待于在具体播放过程中确定，所以这两个事件为非确定性时域事件，它们使得 e_4、s_5、e_5 成为非确定性时域事件。这些非确定性时域事件的存在决定了(b)场景为非确定性时域场景。由于在每次播放同一个多媒体对象的过程中，非确定性时域事件在场景中的位置往往是不相同的，这就意味着表示及处理非确定性时域场景的难度，要比确定性时域场景大得多。

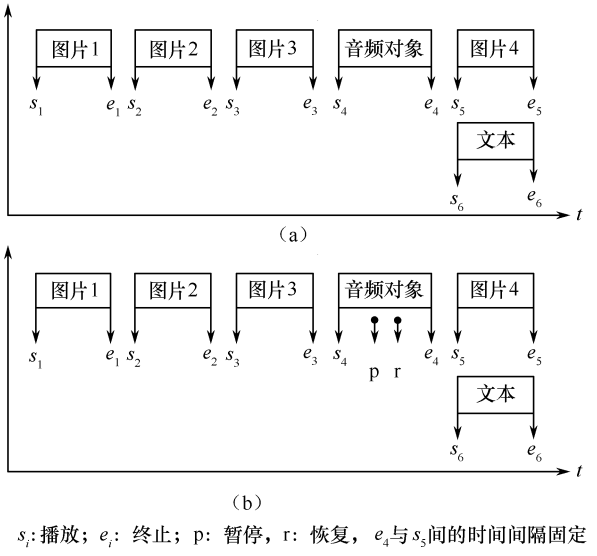


图 6-6 确定性时域场景和非确定性时域场景

在将一个多媒体对象抽象为一个时域场景之后，需要利用某种**时间模型**对场景加以描述。时间模型是一种数据模型，由若干基本部件以及部件的使用规则构成。它是在计算机系统内部为时域场景建模的依据。建模的结果通过某种形式化语言转化为形式化描述，这种形式化描述就是同步描述数据。时间模型及相应的形式化语言则合称为**时域定义方案**(Temporal Specification Scheme)[3]。

为了使同步机制能够了解并维持多媒体对象的时域特征，除了同步描述数据以外，还需要向同

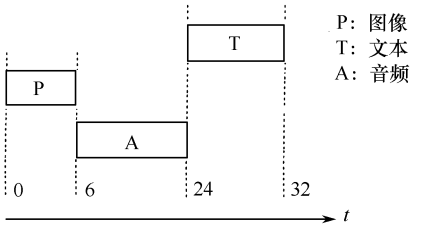


图 6-7　一个时域场景

步机制提出必要的服务质量要求，这种要求是用户和同步机制之间，在应当以何种准确程度来维持时域特征方面所达成的一种约定。这种约定就是我们在前面所说的同步容限。

需要指出，在对同一时域场景进行表示（描述）的过程中，采用的时间模型不同，得到的同步描述数据也就不完全相同。如对图 6-7 所示的时域场景而言，可以得到如下 3 种同步描述数据：

（1）时间模型针对事件发生的时刻

$t=0$ 秒时显示图像 P，$t=6$ 秒时停止显示图像 P；

$t=6$ 秒时播放音频 A，$t=24$ 秒时停止播放音频 A；

$t=24$ 秒时显示文本 T，$t=32$ 秒时停止显示文本 T。

（2）时间模型针对事件发生的相对时刻

音频 A 的播放时刻比图像 P 的显示时刻晚 6 秒；

文本 T 的显示时刻比图像 P 的显示时刻晚 24 秒；

文本 T 的显示时刻比音频 A 的播放时刻晚 18 秒。

（3）时间模型针对事件对应的时间间隔

图像 P 的显示间隔为 6 秒，音频 A 的播放间隔为 18 秒，文本 T 的显示间隔为 8 秒；

A 间隔紧接着 P 间隔，T 间隔紧接着 A 间隔；

T 间隔与 P 间隔相差 24 秒（以间隔起点计）。

模型（1）直接定义事件在场景中的位置，事件间的时域关系间接地得到体现；而模型（2）、（3）则直接对事件间的时域关系进行定义。3 种时间模型从不同的角度来表示同一时域场景，得到不同的同步描述数据。它们都能全面地反映该场景的时域特征，是等效的。但是不同的时间模型具有各自的优缺点，在实际应用中，应当根据系统所要处理的多媒体数据的特点及其应用的需要，选择恰当的时间模型来完成对其时域特征的表示。有时还需要对不同的时间模型进行综合，或提出新的时间模型。对时间模型的分析、比较、综合以及提出新的时间模型等方面的工作，需要在一个统一的基础上进行，下面要介绍的时域参考框架[3]就是这样一个基础。

6.2.2　时域参考框架

时域参考框架（如图 6-8 所示）是研究多媒体同步的一个很好的基础。它不仅有助于分析、比较现存的各种时间模型的优缺点，也为综合不同模型的优点并结合具体应用来定义新的时间模型提供了思路。

时域参考框架由多媒体场景、时域定义方案和同步机制 3 部分构成。多媒体场景是对多媒体数据时、空等方面特征抽象的结果，反映了多媒体数据在这些方面所具备的语义，而时域场景则是多媒体场景的一个重要组成部分，是时域定义方案处理的对象。

如前所述，时域定义方案是在计算机系统内为时域场景建模并对建模结果进行形式化描述的方法，由时间模型和形式语言两部分构成。前者为时域定义方案的语义部分，而后者

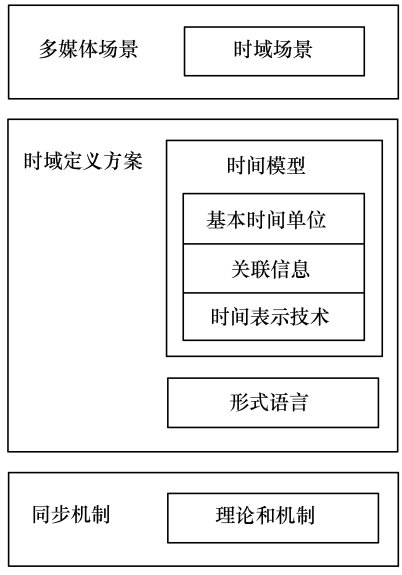


图 6-8　时域参考框架

为其语法部分。通过时域定义方案把时域场景转化为同步描述数据。同步描述数据则是同步机制处理的对象。

同步机制是一种服务过程，它能够了解同步描述数据所定义的时域特征，并根据用户所要求的同步容限，完成对该特征的维护（在处理、传输和运行过程中保证时域特征不遭到破坏）。**流内同步**与**流间同步**是同步机制所要完成的两个主要任务，前者旨在维护连续媒体对象内部的时域关系，后者以保持媒体对象间（连续媒体之间或连续媒体与离散媒体之间）的时域关系为目的。

时域定义方案，即如何描述多媒体数据的时域特征，通常是多媒体著作软件中涉及的问题，这类软件为用户提供定义时域关系和创建多媒体文档的一组工具。由于本书着重于多媒体通信，在多媒体通信系统中，数据的时域特征通常是在获取时建立的（实时同步），无须特别指定，因此，我们仅在 6.2.3 节中简要地介绍描述时域特征的时间模型，而本章余下的篇幅则主要讨论维护时域特征的同步机制。

6.2.3 描述时域特征的时间模型

如图 6-8 所示，一个时间模型由**基本时间单位**、**关联信息**（Contextual information）和**时间表示技术**（Time representation techniques）3 部分构成。

1. 基本时间单位

基本时间单位可以分为**时刻**和**间隔**两种类型。时刻是持续时间为零的瞬间。间隔的定义如下[4]：

令$(S,\leqslant)$表示一个偏序集合，a、$b\in S$ 且 $a\leqslant b$，集合 $\{x \mid a \leqslant x \leqslant b\}$ 则称为间隔，由$[a,b]$表示。$[a,b]$有如下特性：

(1) $[a,b]=[c,d] \Leftrightarrow a = c$ 且 $b = d$

(2) 若 c、$d \in [a,b]$，$e \in S$ 且 $c \leqslant e \leqslant d$ 那么 $e \in [a,b]$

(3) 间隔长度 $\sharp([a,b]) \geqslant 1$

2. 关联信息

时间模型在利用基本时间单位表示时域事件的同时，利用**关联信息**反映时域事件的组织方式。关联信息可以分为定量关联信息和定性关联信息两类。

采用**定量关联信息**的时间模型，认为场景中的各时域事件是相互独立的，因此可以单独地描述每一个时域事件在场景中的位置，从而间接地反映事件间的关系。通过某种计时单位将事件的位置信息表示为数量，如 $t_1 = 6$ pm，$\sharp([a,b]) = 3$ 小时等。这些描述事件位置的数量则构成了定量关联信息的内容。

采用**定性关联信息**的时间模型，认为场景中的各时域事件彼此相关，因此在其关联信息中包含的是对时域事件约束关系的描述（如时刻 A 在时刻 B 后第 3 秒，间隔 A 在间隔 B 之前 8 秒等）。定性关联信息可以是对各时域事件发生次序的描述，称为**次序信息**。除了次序信息以外，有些时间模型的定性关联信息包含了对事件间时域关系的描述。可进一步分为：

(1) 两个时刻之间的基本时域关系

两个时刻之间的基本时域关系包括 before、after 和 at_the_same_time（如图 6-9 所示）。对于确定性时域场景而言，任意两个时刻之间仅存在上述 3 种基本时域关系中的 1 种。

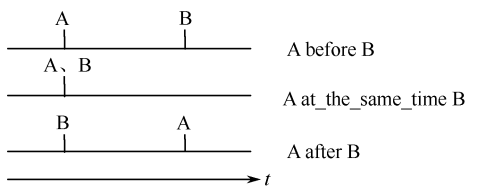


图 6-9 两个时刻之间的基本时域关系

(2) 两个间隔之间的基本时域关系

两个间隔之间有 13 种时域关系[4]，其中 6 种关系

可由其他关系的逆来表示，例如 after 可看成 before 的逆，表示为 before^{-1}。由于 equals 的逆与 equals是等价的，因而我们只需要研究 13 种时域关系中的 7 种，即 before、meets、overlaps、during^{-1}、starts、finishes^{-1}和 equals。这些时域关系可由图 6-10 表示。对于确定性时域场景，任意两个间隔之间仅存在一种基本的时域关系，并且可由 τ_α、τ_β、τ_δ和 τ_{TR} 这 4 个数值参数唯一地确定（见图 6-10）。

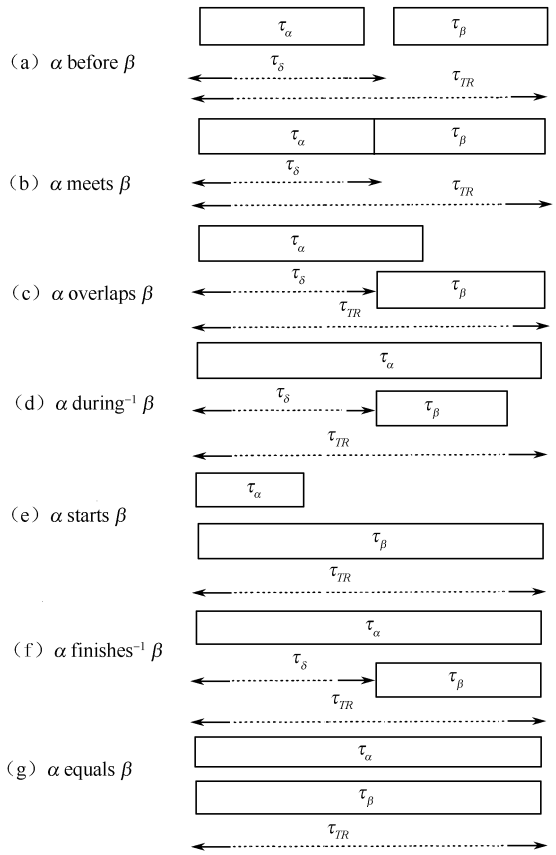


图 6-10　两个间隔之间的时域关系

（3）非确定性时域关系

非确定性时域关系是指那些在基本时间单位之间存在的、不能够用（1）、（2）中所讲的基本时域关系来明确描述的时域关系。这种非确定性的时域关系可以用基本时域关系的析取模式（Disjunctive Normal Form）来表示，如时刻 A{before ∨ after}时刻 B 等[5]。

3. 时间表示技术

时间表示技术是时间模型的第三个组成部分，是时间模型依照关联信息来定义场景中各事件与时间轴之间对应关系的方法，是多媒体对象的播放、传送等机制根据描述数据生成调度方案的出发点。我们将通过下面介绍的几种典型的时间模型，对时间表示技术作进一步的说明。

4. 典型的时间模型

（1）时间轴模型

时间轴模型[6]的基本时间单位为时刻，其关联信息为定量关联信息。这是一种基本的、容易理解的对时域场景的建模方法，其定量关联信息包含的是事件发生的准确时间。在具体的建模过程中，媒体对象的开始播放、终止播放等事件首先被抽象为时刻，然后通过定量关联信息，将每个时域事件在场景中的位置独立地确定下来。时间表示技术则根据定量关联信息（事件发生的时刻），直接建立起各时刻与一个全局时间轴的对应关系。在运行时，系统则按这个全局时间轴规定的动作执行。图 6-12 给出了针对图 6-11 所示的时域场景，依据时间轴模型所得到的建模结果，图中的 τ 表示媒体对象的时间宽度。该场景是演示某一应用软件，首先播放的图像/伴音片段介绍该软件的特色，RI 代表事先采录好的用户对软件的使用情况，P_1、P_2 和 P_3 三张图片反映使用的结果，动画及第二段音频片段进一步解释软件的使用方法。与此同时，允许用户在仿真环境中对软件进行实际操作（UI），用户交互操作结束后显示图片 P_4。

由于时间轴模型认为场景中各时域事件是相互独立的，所以向场景中添加或从场景中删除一个时域事件并不影响对其他时域事件的描述，这就大大降低了维护描述数据的复杂程度。图 6-12 中的问号“?”表明，由于时间轴模型的定量关联信息包含的是事件发生的准确时间，因而该模型难于用来表示非确定事件（如图 6-11 中的 UI）。这是时间轴模型的一个缺点。

（2）虚时间轴模型

虚轴模型[7]的基本时间单位为时刻，关联信息为定性关联信息（次序信息）。该模型是对时间

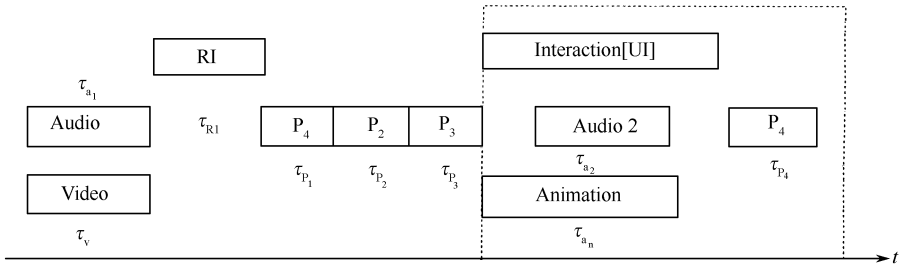


图 6-11　典型时域场景

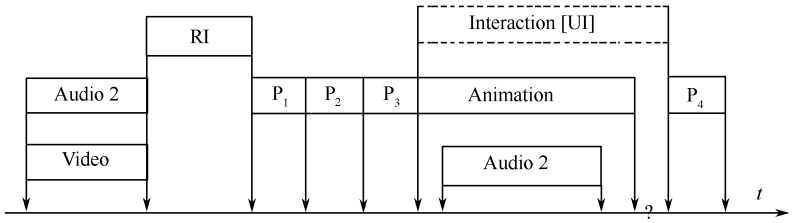


图 6-12　时间轴模型与典型时域场景

轴模型的一种扩展，它一方面继承了时间轴模型简单、同步描述数据易于维护的优点，又具备了较强的表示非确定性时域场景的能力。

在具体的建模过程中，可以引入多条时间轴。时间轴的计时单位可以是物理计时单位（如毫秒、秒、时钟脉冲等），这样的时间轴称为物理时间轴；也可以是逻辑计时单位（如事件发生次序的编号等），这样的时间轴称为逻辑时间轴。时域场景中的确定性事件映射到物理时间轴上，非确定性时域事件则根据次序信息映射到逻辑时间轴上。图 6-13 给出了针对图 6-11 所示的时域场景、依据虚轴模型所得到的建模结果。在系统运行过程中，当用户操作时（UI 发生的时间确定），才将逻辑时间轴上的事件（UI，P_4）映射到物理时间轴上。由于采用了多条时间轴，所以由虚轴模型得到的描述数据较为复杂，对其进行维护的难度要略高于时间轴模型。

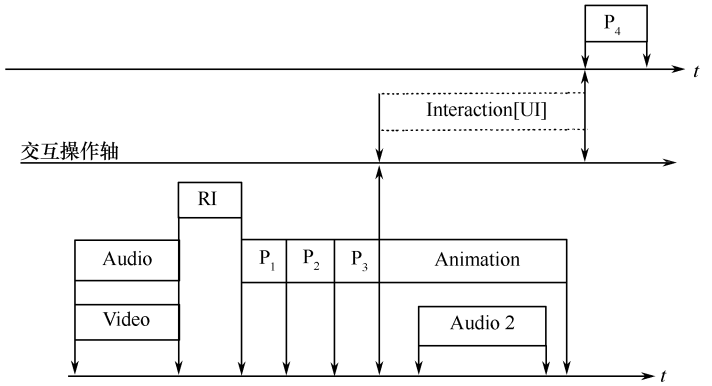


图 6-13　虚轴模型与典型时域场景

（3）OCPN 模型

OCPN（Object Composition Petri Net）**模型**[8]的基本时间单位为间隔，关联信息为定性关联信息，其定性关联信息包含的是对两个间隔之间基本时域关系的描述，且为 τ_α、τ_β、τ_δ、τ_{TR} 等时间参数进一步限定。实际上，OCPN 模型是对 Petri Net 模型的一种拓展，Petri Net 是一种二分有向图，可表示为 $N=(T,P,A)$，其中：T、P、A 分别表示转移点集、位置集和有向弧集。拓展后得到的

OCPN 模型则可表示为 $C_{\text{O CPN}}=\{T,P,A,D,R,M\}$，其中 T、P、A 的含义不变，模型的基本时间单位（即间隔）对应于位置，而 D、R、M 则标识了位置的属性，其中 D 表示各位置的持续时间，R 表示各位置所需的资源，M 则表示各位置所含令牌的个数。

图 6-14 给出了根据 O CPN 模型对两个间隔之间各种基本时域关系的表示结果，其中的竖线表示转移点。针对图 6-11 所示的时域场景，图 6-15 给出了依据 OCPN 模型所得到的建模结果。图中位置 P_δ 是为了表示 Audio 2 和 Animation 之间的 During 关系而引入的，τ_δ 是 Audio 2 的起始时刻对应于 Animation 的起始时刻的延时。

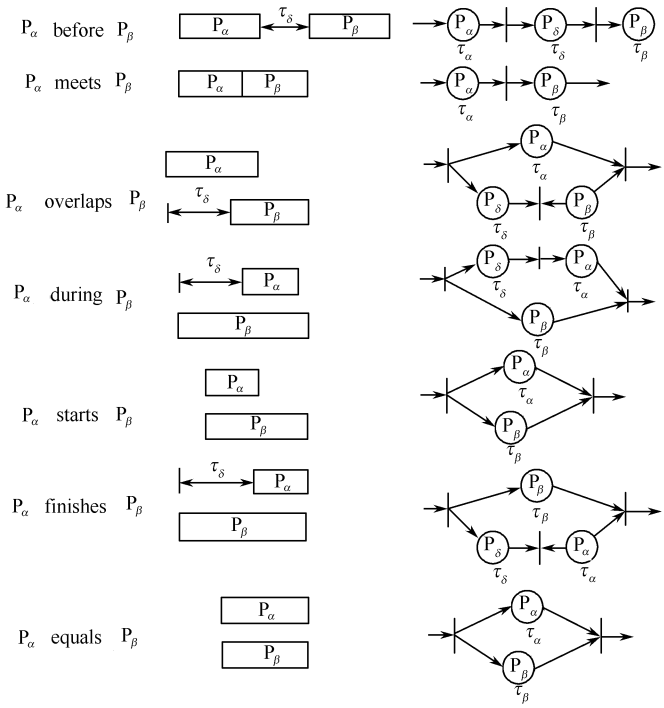


图 6-14　两个间隔之间基本时域关系的 OCPN 表示

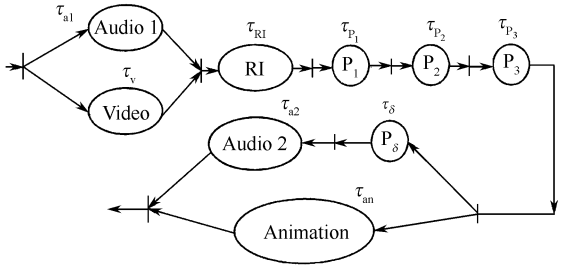


图 6-15　OCPN 模型与典型时域场景

OCPN 模型的时间表示技术称为**点火原则**（Firing Rules），它定义了依据建模结果恢复时域场景的方法。OCPN 模型的点火原则为：

① 当转移点 τ_i 的所有输入位置都含有一个未锁定的令牌时，转移点点火；

② 点火后，转移点 τ_i 从它的每个输入位置上移走一个令牌，并给它的每个输出位置加入一个令牌；

③ 位置 P_j 收到令牌后，在时间间隔 τ_j 内保持活跃状态（执行该时域事件），且令牌被锁定；经过时间 τ_j 后，位置 P_j 进入不活跃状态，同时令牌被解锁。

从以上分析可以看出，基本的 OCPN 模型不具备表示非确定性时域场景的能力，但近年来对该模型的扩展已使其具备了这种能力。此外，由于 OCPN 模型直接描述媒体对象间的时域关系，由它得到的描述数据较为复杂，不宜维护，但是点火原则很方便于同步机制实施同步控制。

6.2.4 同步容限

如 6.1.3 节所述，对多媒体数据时域特征的表示包含同步描述数据和同步容限两个部分，前者决定了多媒体数据在时域中的布局，而后者则包含了对同步机制服务质量的要求。二者结合起来称为同步规范。

在一个多媒体系统的实际运作过程中，总存在着一些妨碍准确恢复时域场景的因素，例如其他进程对 CPU 的抢占、缓冲区不够大、传输带宽不足等，这些因素往往导致在恢复后的时域场景中，时域事件间的相对位置发生变化（如图 6-16 所示）。我们将这种变化称为**事件间偏差**（Skew）。属于同一媒体对象的时域事件之间的偏差称为**对象内偏差**，不同媒体对象的时域事件之间的偏差为**对象间偏差**。偏差的存在必然会造成多媒体同步质量的降低，而同步容限则包含了用户对偏差许可范围的定义。同步机制需依据同步容限，保证在恢复后的时域场景中，事件间的偏差在其许可范围之内。

对多媒体同步质量的评估方式，直接影响着用户对偏差许可范围的规定。由于很难找到定义偏差许可范围的客观标准，通常采用的办法是主观评估。虽然由主观评估所得到的偏差许可范围并不是十分准确，但仍可作为设计多媒体同步控制机制的参照。对于对象内的偏差，人们能够察觉 30 ms 左右的图像信号的不连续性，而一个连续的单音调的声音信号间断 1 ms 便能察觉得到。对于对象间的偏差，表 6-2 给出了由主观评估所得到的大致许可范围[1]。

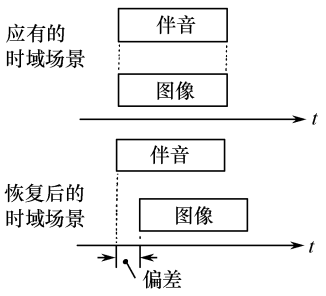


图 6-16 事件间的偏差

表 6-2 媒体间偏差的许可范围

媒体		条件	许可范围
视频	动画	相关	±120 ms
	音频	Lip-syn	±80 ms
	图像	重叠显示	±240 ms
		不重叠显示	±500 ms
	文本	重叠显示	±240 ms
		不重叠显示	±500 ms
音频	音频	紧密耦合（立体声）	±11 μs
		宽松耦合（会议中来自不同参加者的声音）	±120 ms
		宽松耦合（背景音乐）	±500 ms
	图像	紧密耦合（音乐与乐谱）	±5 ms
		宽松耦合（幻灯片）	±500 ms
	文本	字幕	±240 ms

6.3 分布式多媒体系统中的同步

我们将信息获取、处理、存储和播放都在一台多媒体计算机中进行的系统称为单机系统，而信息的提供者（信源）和接收者（信宿）相处异地、需要由网络相连接的系统称为**分布式**多媒体系统。虽然在单机系统中也存在媒体同步的问题，但是在分布式系统中的同步问题则更为重要。

6.3.1 分布式多媒体系统的结构

图 6-17 给出了分布式多媒体系统可能具有的结构。图（a）所示的是只有一个信源和一个信宿

的情况，称为点对点结构。可视电话是这种结构的一个典型应用；(b)是一点对多点结构，由一个信源向多个信宿发送信息。远程教学、IPTV 等应用都属于这种情况。(c)为一个信宿、多个信源的结构，这可以是一个用户从分布式数据库中得到查询的结果，例如从一个数据库中得到视频信息，而从另一个库中获得相关的音频信息。图(d)表示的是上述例子可能具有的其他结构，即先将从分布式数据库中提取出的相关联的查询结果集中到一个中间点，再将复合后的多媒体信息送至用户。图(e)则是多点对多点的情况。在这里用户构成了一个组，组内的用户可以为信源，也可以为信宿，或者既是信源又是信宿，多媒体会议就是这种结构的典型例子。

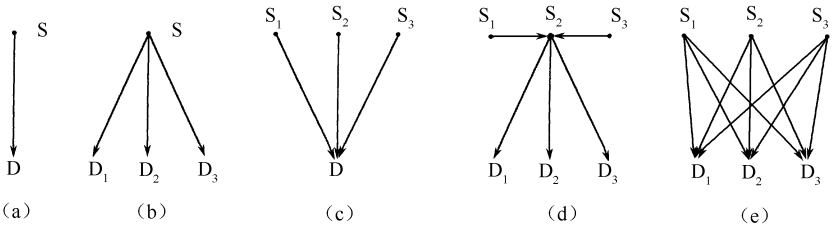


图 6-17　分布式多媒体系统的结构

在有多个信源和(或者)多个信宿的情况下，多媒体同步除了要考虑媒体内部和媒体之间的同步外，还存在一个特殊的问题，那就是从各个信源发出的信息是否同步地到达各个信宿，这通常称为**组同步**，或群同步。

6.3.2　同步规范的传送

由于在分布式多媒体系统中信源与信宿是分离的，而信宿在播放某个多媒体对象之前，必须已知与之相关的同步规范才能进行播放，这就提出了在从信源向信宿传送多媒体数据的同时，如何传送相关的同步规范的问题。可能采用的方式有如下几种：

(1) 在传送多媒体数据之前，将整个播放过程所需要的全部同步规范，以文件方式传送到接收端，以备播放时参照。这种方式简单易行，特别是在有 n 个信源的情况下优点更为明显。但是这种方法要引入一定的附加延时，数据的传输只能在同步规范传送之后才能进行。而且这种方式只适用于综合时域同步关系，对在通信过程中实时产生的同步关系是不可行的。

(2) 在传送多媒体成分数据的通道之外，使用一个附加的逻辑通道专门传送同步规范(在第 7 章中我们将会看到，有的网络可以支持多个逻辑通道)。这种方法主要适用于系统中只有一个信源，而且时域同步关系是实时产生、无法预先知道的情况。使用这种方法时要注意，必须保证在多媒体数据的播放过程中，经另一个通道传送的对应的同步规范能按时、无误地到达接收端，否则播放将因缺乏相应的时域关系而中断。

(3) 将同步规范与成分数据复接在一起，使用一个通道传送。这是最为广泛使用的一种方法，其优点是没有附加的延时，也不需要附加另一个通道。在许多传送音视频的多媒体系统中(见 8.2.1 和 8.2.2 节)，同步(时间关系)信息都是插入在视、音频数据中经同一个通道传送的。

6.3.3　影响多媒体同步的因素

在分布式多媒体系统中，信源产生的多媒体数据需要经过一段距离的传输才能到达信宿。在传输过程中，由于受到某些因素的影响，多媒体数据的时域约束关系可能被破坏，从而导致多媒体数据不能正确地播放。下面将可能影响多媒体同步的因素，分别叙述如下：

1. 延时抖动

信号从一点传输到另一点所经历的延时的变化称为**延时抖动**。系统的很多部分都可能产生延

时抖动。例如从磁盘中提取多媒体数据时，由于存储位置不同导致磁头寻道时间的差异，各数据块经历的提取延时有所不同；在终端中，由于CPU、存储单元等资源的不足可能导致对不同数据块所用的处理时间不等；在网络传输方面也存在着许多因素使信源到信宿的传输延时出现抖动，7.2.1节将对此作详细的分析。

延时抖动将破坏实时媒体内部和媒体之间的同步。图6-18给出了网络延时抖动对同步破坏的例子。在信源端，视频流和音频流内各自的LDU之间是等时间间隔的，两个流的LDU之间在时间上也是对应的。在信宿端，由于各个LDU经历的传输延时不同，视频流和音频流内部LDU的时序关系出现了不连续，二者之间的对应关系也被破坏。

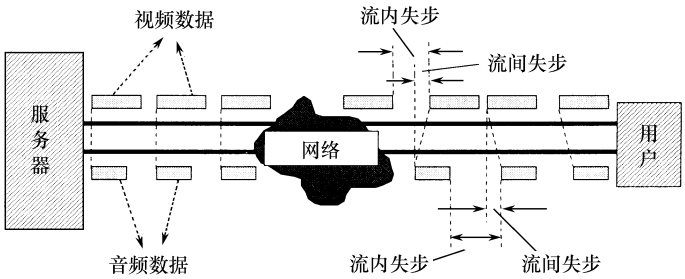


图6-18　传输延时抖动对多媒体同步的破坏

2. 时钟频率偏差

在无全局时钟的情况下，分布式多媒体系统的信源和信宿的时钟频率可能存在着偏差。多媒体数据的传送是基于发送时钟进行的，而它的播放则是由信宿端的本地时钟驱动的，如果信宿的时钟频率高于信源的时钟频率，经过一段时间后可能在收端产生数据不足的现象，从而使连续媒体播放产生停顿；反之，则可能使收端缓存器溢出，图6-19描述了这两种情况。图中从原点开始的直线表示LDU的发送时刻，箭头的长度表示每个LDU的传输延时，T为播放的起始延时，从T开始的直线表示LDU开始播放的时刻。两条直线的斜率差反映了收、发时钟的频差，深色区域代表LDU在缓存器内停留的时间。从图(a)看出，由于接收时钟频率低于发送端，一段时间后缓存器会溢出。当接收时钟频率高于发送端时，从图(b)看出，一段时间后播放时刻已经早于LDU的到达时刻(缓存器变空)。

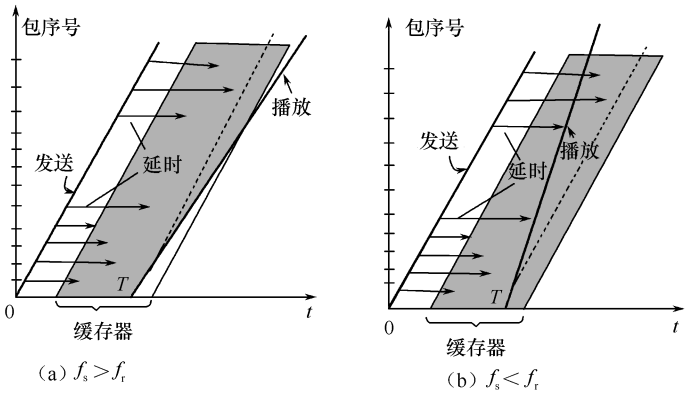


图6-19　收发时钟频率偏差对多媒体同步的影响

需要注意，即使在收、发时钟标称频率相同的情况下，二者之间的偏差还是有可能存在的。这种偏差可能由于收、发时钟的实际频率不精确地等于其标称频率(精确度问题)；或者由于电源电压、温度等因素影响，使其中一个或两个时钟的频率发生了变化(稳定度问题)。

3. 不同的采集起始时间或不同的延时时间

在多个信源的情况下[如图 6-17(c)和(e)],信源必须同时采集和传输信息。例如一个信源采集图像信号,另一个采集相关联的伴音信号,如果二者的采集的起始时间不同,在接收端同时播放这两个信源送来的媒体单元必然出现同步的问题。两个信源到信宿的传输延时不等或者打包/拆包、缓存等时间的不同,也会引起同样的问题。又如会议多个参与者的信号需要在某个中间节点混合成一个信号,传送给其他(或所有)参与者,如果参与者的发送起始时间不同或传输延时不同,在信宿端则得不到按正确时间关系混合的信号。

4. 不同的播放起始时间

在有多个信宿的情况下[见图 6-17(b)和(e)],各信宿的播放起始时间应该相同。在某些应用中,公平性是很重要的。如果用户播放的起始时间不同,获得信息早的用户较早地对该信息作出响应,这对其他用户就是不公平的了。以上 3 和 4 两个因素通常为组同步中存在的问题。

5. 数据丢失

传输过程中数据的丢失相当于该数据单元没有按时到达播放器,显然会破坏同步。

6. 网络传输条件的变化

我们在第 7 章中将会看到,在一些重要的网络上,例如 IP 网等,网络的传输延时、数据的丢失率均与网络的负载有关,因此在通信起始时已经同步的数据流,经过一段时间后可能因网络条件的变化而失去同步。

6.3.4 多级同步机制

在实际的多媒体系统中,同步机制往往不是作为一个独立的模块存在的,而是分散于系统的各个部分,即每个部分都要采取自己的同步措施,分别保证各自的同步关系。这些部分包括:

(1) 采集多媒体数据及存储多媒体数据时的同步;

(2) 从存储设备中提取多媒体数据时的同步;

(3) 发送多媒体数据时的同步;

(4) 多媒体数据在传输过程中的同步;

(5) 接收多媒体数据时的同步;

(6) 各类输出设备内部的同步。

其中(3)~(5),即发送、传输和接收过程中的同步控制是分布式多媒体系统中特有、而单机系统中没有的,可以总称为**多媒体通信的同步机制**,这将是我们在以后几节讨论的重点。值得指出,本章所讨论的有关同步的方法和思想也可以推广应用到(1)、(2)和(6)中。

在同步机制中,通常首先进行连续媒体的流内同步;然后进行流间同步;最后再考虑连续媒体和静态媒体之间的同步。由于静态媒体对象自身没有时间特征,而且静态媒体对象与连续媒体对象之间的同步容限又较宽松,两者间的同步控制比较容易实现,所以在多媒体通信中,静态媒体对象的传输以及静态媒体对象和连续媒体对象间的同步控制并不是需要解决的主要问题。我们在下面几节中,将主要对连续媒体的流内和流间的同步控制以及收、发时钟的同步问题进行介绍。

6.4 连续媒体内部的同步

6.4.1 基于播放时限的同步方法

连续媒体数据是一个由 LDU 构成的时间序列,LDU 之间存在着固定的时间关系。如图 6-18

表示出，当网络传输存在延时抖动时，连续媒体内部 LDU 的相互时间间隔会发生变化。这时，在接收端必须采取一定的措施，恢复原来的时间约束关系。一个最简单的办法是让接收到的 LDU 先进入一个缓存器（见图 6-20），对延时抖动进行过滤，让从缓存器向播放器（或解码器）输出的 LDU 序列是一个连续的流。如果缓冲器的容量为无穷大，相当于把整个数据流全部传送到接收端之后再进行播放，显然无论多么大的延时抖动都可以滤除掉，但这时的起始播放延时却可能达到不可容忍的程度。此外，如果数据流是在通信过程中实时产生的（如可视电话），这种方法也是不可行的。因此，必须适当地设计缓冲器的容量，使它既能消除延时抖动的影响，又不过分地加大起始播放的延时时间。

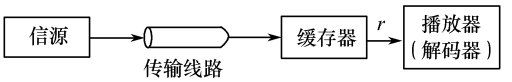


图 6-20　接收缓存器

假设发送端时钟和接收端（播放）时钟是同步的，并且在发送端实时媒体内部是同步的，即各个 LDU 的发送时间间隔为一常数。若第 i 个 LDU 的发送时刻为 $t(i)$，则其到达接收端的时刻 $a(i)$ 为

$$a(i) = t(i) + d(i) \tag{6-1}$$

式中，$d(i)$ 为第 i 个 LDU 的传输延时。为了分析的简单，假设延时抖动限定在一个范围之内，即

$$d_{\min} \leqslant d(i) \leqslant d_{\max} \tag{6-2}$$

要保证播放的不间断，第 i 个 LDU 的播放时刻 $p(i)$ 必须晚于它的到达时刻 $a(i)$，即

$$p(i) \geqslant a(i) \qquad (i = 1,2,\cdots) \tag{6-3}$$

这就是说，$p(i)$ 规定了第 i 个 LDU 到达的最后期限。

由于播放过程必须保持数据内部原有的（在发送端的）时间约束关系，所以在信源和信宿本地时钟是同步的假设下，每个 LDU 有如下关系：

$$p(i) - p(i-1) = t(i) - t(i-1) \qquad (i = 2,3,\cdots) \tag{6-4}$$

即

$$p(i) - p(1) = t(i) - t(1) \qquad (i = 2,3,\cdots) \tag{6-5}$$

其中 $t(1)$ 和 $p(1)$ 分别表示第一个 LDU 的发送和播放时刻。由(6-1)式和(6-5)式得到

$$p(i) - a(i) = p(1) - a(1) - [d(i) - d(1)] \qquad (i = 2,3,\cdots) \tag{6-6}$$

根据(6-3)式，上式可转化为

$$p(1) - a(1) \geqslant [d(i) - d(1)] \qquad (i = 2,3,\cdots) \tag{6-7}$$

在**最坏情况**下，保证上式成立的条件是[9]

$$p(1) - a(1) = \mathrm{Max}\{[d(i) - d(1)] \mid i \in (2,3,\cdots)\} = d_{\max} - d_{\min} \tag{6-8}$$

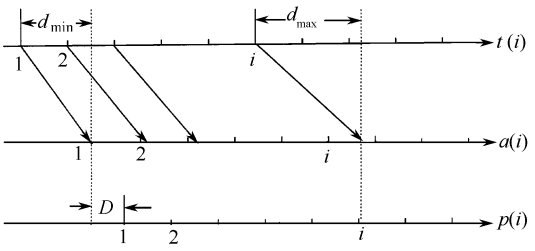


图 6-21　实时数据的发送、接收和播放时间关系

(6-8) 式说明，在延时抖动限定在一定范围的条件下，接收端在接收到第 1 个 LDU 之后，必须推迟一段时间 $D = (d_{\max} - d_{\min})$ 再开始播放，才能保持整个播放过程不间断。时间 D 通常称为**起始时刻偏移量**，或**起始延时**。图 6-21 给出了上述情况的示意图。图中各 LDU 的发送时刻是等间距的，其接收时刻用倾斜的箭头指出。考虑可能发生的最坏情况是，第 1 个 LDU 的延迟时间最小（如图所示）。如果接收到第 1 个 LDU 时就立即播放，那么对于任何一个传输延时大于 $d_{\min}$ 的 LDU，当需要播放它时它都没有到达。如果将播放的起始时间定在 D，则延时最大的第 i 个 LDU 的播放时刻正好与它的到达时刻相同，就消除了播放的不连续。

值得注意，接收端缓存器不仅能够滤除延时抖动的影响（保证缓存器不变空），还要保证在任何情况下不发生溢出。现在根据这一要求来推导缓存器的最大容量。

第 i 个 LDU 在缓存器中缓存的时间为 $[p(i)-a(i)]$，由(6-6)式和(6-8)式得到

$$\begin{aligned} B_t &= \mathrm{Max}\{[p(i)-a(i)] \mid i \in (2,3,\cdots)\} \\ &= (d_{\max}-d_{\min})-\min\{[d(i)-d(1)] \mid i \in (2,3,\cdots)\} \end{aligned} \tag{6-9}$$

式中，B_t 为 LDU 的最大缓存时间。当 $d(1)=d_{\max}$，$d(i)=d_{\min}$ 时，$[d(i)-d(1)]$ 具有最小值，因此缓存器的最大容量 B 为

$$B=\lceil B_t \cdot r \rceil=\lceil 2(d_{\max}-d_{\min}) \cdot r \rceil \tag{6-10}$$

式中，$\lceil\ \rceil$ 为取整，r 为播放速率，它的单位为每秒钟传输的 LDU 的个数。对于变比特率(VBR)的媒体流，如何将以媒体单元为单位的缓存器容量转化为字节数是一个需要慎重处理的问题。对于常速率(CBR)媒体流，原则上讲转化是很简单的，但如果缓存容量只有几帧，而以 H.26X 或 MPEG 等方法压缩的视频流每个 LDU(1 帧)的数据量很不相同，此时如何有效地进行单位转化也是需要注意的。由于同步只与时间有关，对于连续媒体，时间可以间接地用 LDU 的个数表示，因此在讨论同步问题时，我们将只以时间或媒体单元的个数来表示缓存器的容量。

以上介绍的是解决传输延时抖动的方法，由数据提取、数据处理等原因引起的延时抖动问题也可以用类似的方法来解决。

6.4.2 基于缓存数据量的同步方法

上节讲述的方法适用于收发时钟同步、且延时抖动在一个确定的范围之内的情况。本节将介绍一种在更一般情况下使用的方法。

图 6-22 给出了采用这种同步方法的两种系统模型，每一种都包括一个控制环路，其区别在于环路是否将信源和传输线路包含在内。与 6.4.1 节中的方法相似，信宿端也有一个缓存器。缓存器的输出按本地时钟的节拍连续地向播放器提供媒体数据单元，缓存器的输入速率则由信源时钟、传输延时抖动等因素决定。由于信源和信宿时钟的频率偏差、传输延时抖动或网络传输条件变化等影响，缓存器中的数据量是变化的，因此要周期性地检测缓存的数据量，如果缓存量超过预定的警界线，例如快要溢出，或者快要变空，就认为存在不同步的现象，需要采取步骤进行**再同步**[10]。在图(a)中，再同步在信宿端进行，可以通过加快或放慢信宿时钟频率，也可以删去或复制缓存器中的某些数据单元，使缓存器中的数据量逐渐恢复到警界范围之内的正常水平。在图(b)中，类似的再同步措施是在信源端进行。在需要进行再同步时，通过网络向信源反馈有关的控制信息，让信源加快或放慢自己的发送速率。

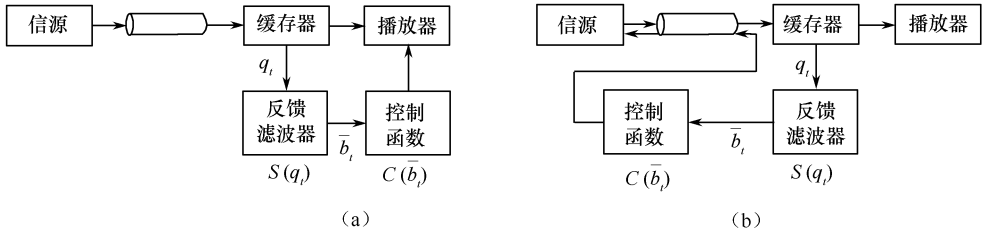


图 6-22 基于缓存数据量控制的系统模型

现在具体讨论环路的工作原理。设 t 时刻的缓存数据量为 q_t，通过环路滤波器 $S(q_t)$ 得到平滑后的缓存数据量 $\bar{b}_t$。典型的环路滤波器采用几何加权平滑函数：

$$\bar{b}_t = S(q_t) = \alpha \cdot \bar{b}_{t-1} + (1-\alpha)q_t \tag{6-11}$$

式中，$\alpha \in [0,1]$ 为平滑因子。上述环路滤波器实际上是一个一阶低通滤波器，由(6-11)式很容易得到它的 Z 变换表达式，要保证滤波器稳定，需要满足 $\alpha \leqslant 1$。在环路中使用低通滤波器的目的是，将由短期(高频)延时抖动而引起的缓存数据量的波动平滑掉，只有由时钟频率偏移、网络传输

条件改变等因素导致的长期(低频)缓存量变化才会触发控制函数 $C(\bar{b}_t)$ 的工作。α 的数值直接影响控制的灵敏度。如果 α 过大,再同步的控制机制启动太缓慢,可能导致缓存器的变空、或溢出;如果 α 过小,延时抖动引起的缓存量的微小变化也会不必要地启动再同步机制,导致系统的不稳定或振荡。

控制函数 $C(\bar{b}_t)$ 将 $\bar{b}_t$ 与预先设定的缓存量警界线相比较(见图 6-23),在正常情况下,$\bar{b}_t$ 在上警界线 UW 和下警界线 LW 之间浮动。如果 $\bar{b}_t > UW$、或者 $\bar{b}_t < LW$,则分别表示缓存器有溢出、或者变空的危险,必须启动再同步机制。这时以图 6-22(a)所示系统为例,信宿可参照下式调整自己的播放速率[11]:

$$R' = R(1 + R_C) \tag{6-12}$$

$$R_C = \frac{\bar{b}_t - [LW + (UW - LW)/2]}{L} \tag{6-13}$$

式中,R 为正常播放速率,R_C 为相对调整比率,L 为再同步的调整期(即在该段时间内改变播放速率)。在调整期 L 结束时,$C(\bar{b}_t)$ 检查 $\bar{b}_t$ 是否回到正常水平,如果是,不再进行调整;如果 $\bar{b}_t$ 仍在警界线之外,则再启动一个新的再同步调整期。显然,在调整期 L 内,播放的连续性会受到一定程度的破坏,特别是音频速率较大的变化将使人耳感觉到音调变化。另一种调整的方法是通过删去或重复缓存器中的数据单元来实现再同步。在每一个调整期内,删去或重复的数据量可以是一个固定值,也可以是一个变化值,该值正比于 $\bar{b}_t$ 超出警界线的数据量。如果一次调整不能使 $\bar{b}_t$ 回到正常水平,则再启动一个新的调整周期。此种调整方法不引起声音音调的变化,只会出现轻微咯咯声;而对视频,则会出现画面的跳动或暂停。

现在来讨论图 6-23 所示的缓存器的容量。(6-10)式给出了在延时 $d \in [d_{\min}, d_{\max}]$ 条件下,保证媒体内部同步所需的缓存器容量 B。在图 6-23 中,用 B 作为 LW 和 UW 之间的容量。这意味着,当延时 d 在预先选定的范围 $[d_{\min}, d_{\max}]$ 内时,其抖动可以通过缓存 B 得到补偿,流内同步能够得到保证;当 d 超过上述范围时,就需要启动再同步机制。由于反馈滤波器的作用,q_t 的变化反映到 $\bar{b}_t$ 的变化需要经过一段时间 τ。例如,q_t 已经高于 UW,$\bar{b}_t$ 则还需要经过 τ 时间后才会高于 UW。我们称 q_t 超过警界线的时刻为产生不同步的时刻,而 $\bar{b}_t$ 超过警界线的时刻为 $C(\bar{b}_t)$ 检测到失步的时刻。图 6-23 中附加的缓存器容量 b_a(以时间计算)至少应该足够容纳这一段时间之内的数据,否则在 $C(\bar{b}_t)$ 检测到失步状态之前,缓存器就已经溢出而使数据丢失。如果容量 b_a 能覆盖从失步的产生、检测和重新再同步的整个时间段,便可以使失步对播放质量的影响减至最小。同样,q_t 低于 LW 的情况也可以作类似的分析。值得注意,在缓存器容量如图 6-23 所示的情况下,只有第 1 个 LDU 超过了 LW 之后,才能按(6-8)式规定的时间开始播放。因而 b_a 越大,播放的起始延时时间越长。

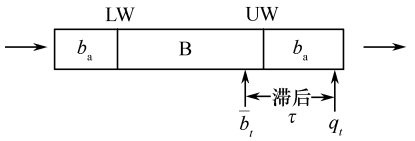


图 6-23 缓存器数据量控制

对于图 6-22(b)所示的系统,由于传输线路被包含在反馈环路之内,使得检测到失步的时刻到再同步调整之间增加了一个传输延时,因此在设计缓存器容量 b_a 时,必须考虑到这个延时。同时应注意,向信源反馈再同步控制信息的时间间隔要长于环路的响应时间(包括上述传输延时),否则容易引起系统的不稳定。

6.5 媒体流之间的同步

媒体之间的同步包括静态媒体与实时媒体之间的同步和实时媒体流之间的同步,由 6.3.4 节中所述的原因,我们将只讨论后者。对于媒体流之间同步的方法没有通用的模式,许多方法都是基

于特定的应用环境而提出的。读者可以通过下面的典型示例，掌握如何实现媒体流之间同步的方法，以便结合自己所遇到的实际问题，加以运用和发展。

6.5.1 基于全局时钟的时间戳方法

假设在图 6-24(a)所示的例子中，有两个信源和一个信宿。所有信源和信宿的本地时钟都与一个全局时钟同步。信源 A 送出的视频数据流和信源 B 送出的音频数据流应该在接收端 C 同步地播放。从 A 到 C 和从 B 到 C 的传输延时分别为 d_1 和 d_2，且网络没有延时抖动。

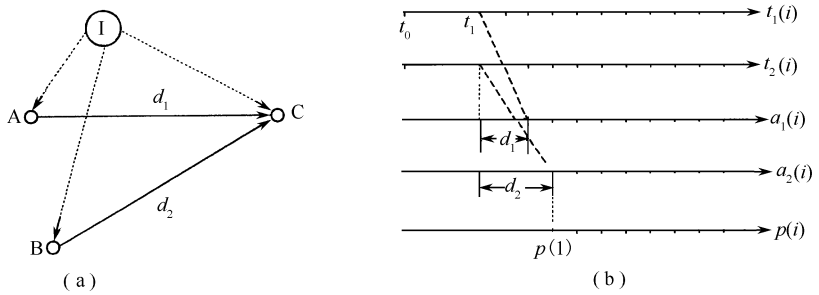


图 6-24 基于全局时钟的时间戳方法

在这个例子中，不需要考虑时钟频率偏差和延时抖动，影响**组同步**的主要因素为两个信源不同的发送时间和不同的传输延时。显然，如果两个信源能在同一时刻，如 t_1，开始传送数据，而接收端从 t_1 开始等待一个最大延时 $max\{d_1, d_2\}$ 后才开始播放，就能达到组同步的目的。这个思想虽然简单，但在分布式系统中实现却并不容易。下面给出的做法是使用一个称为**启动器**的处理单元来建立起所有成员的共同起始时刻 t_1[12]。

在系统启动时，由启动器 I 向 A、B 和 C 发送有关的控制信息，如参考时刻 t_0、同步区间的起始时间 t_1 等。I 可以安装在任何一个信源或信宿。参考时刻 t_0 必须选在保证所有的信源和信宿都能接收到控制信息之后才开始。从 t_0 到 t_1 的时间[见图 6-24(b)]为同步的预备时间，在这段时间里信宿与同一同步组中的其他信宿(如果同步组中有多个信宿的话)相互交换有关信息，例如信源到本信宿的延时 d_j 等。t_0 到 t_1 的时间也必须足够长以保证这些信息交换的完成。信源从 t_1 开始向外发送 LDU，并根据本地时间给每个 LDU 打上**时间戳**(Timestamp)。由于收、发端已经协商好了共同的参考时刻 t_0，因此它们的计时都以 t_0 为基准。信源发送第 i 个 LDU 的时间截记为 $t_s(i) = [t(i) - t_0]$，其中 $t(i)$ 为该 LDU 发送时刻所对应的本地时钟计数值，t_0 为信源的参考时刻。时间截可以记录在 LDU 的包头中。假设该信源与信宿之间的传输时延为 d，则信宿收到该包的时刻为 $t_0 + t_s(i) + d$。这里的 t_0 是信宿的参考时刻。为了保证从不同信源发送来的第 i 个 LDU 能在信宿同步播放，将各码流的第 i 个 LDU 提交给播放器的时刻 $T(i)$ 应为

$$T(i) = t_0 + t_s(i) + \Delta \tag{6-14}$$

式中，$\Delta = \max\{d_j \mid j \in (1,2,\cdots,n)\}$，$d_j$ 为在 n 个信源和 1 个信宿组成的同步组中，第 j 个信源与信宿之间所需要的传输延时。图 6-24(b)表示出了相应的各个时间关系。

如果在通信过程中延时 $d_j(j=1,2,\cdots,n)$ 发生变化，同步过程可以分为若干个同步区间自适应地进行。启动器 I 在起始时将同步区间的长度发送给同步组内的所有成员。在一个同步区间内，信宿不仅根据(6-14)式进行同步播放，而且将接收到的 LDU 所携带的时间戳与接收到该 LDU 的本地时间作比较，从而得到对当前延时 d_j 的估计值，将此估值在下一个同步预备时间内发送给组内其他信宿。在下一个同步区间开始时，所有信宿将采用当前估值 d_j 来进行(6-14)式所规定的播放调度。

6.5.2 基于反馈的流间同步方法

假设在图 6-25 所示的多媒体信息查询系统中，各用户查询到的媒体数据流需要同步地播放，例如到达用户 A 的音频流和到达用户 B 的视频流的播放必须符合二者之间的同步要求。由于全局时钟的建立需要借助于有关协议和复杂的电路，为了适应更一般的情况，在图 6-25 所示的例子中，我们假设用户以及服务器的时钟都是相互独立的，各用户的播放起始时间也不精确地相同。还假设服务器到每个用户的传输延时都在$[d_{\min}, d_{\max}]$之内，即它们到服务器的距离是大致相等的。

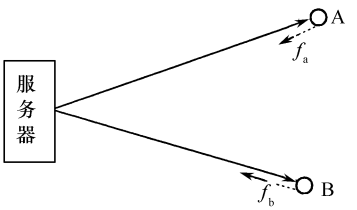


图 6-25 基于反馈的媒体流之间的同步方法

服务器在存储各条媒体数据流时采用的是虚轴模型，即每条流有自己的时间轴，该流的 LDU 按距自己第一个 LDU 起始时刻的相对距离标记时间戳，称为**相对时间戳** RTS(Relative Time Stamp)。在不同的媒体流之间，需要同步播放的 LDU 的 RTS 相同。当向用户传送数据时，服务器根据 RTS 来进行提取和发送调度，不同流中具有相同 RTS 的 LDU 被同时提取出来，并送到相应的通信线路上去。虽然服务器的提取和发送是同步的，但由于各用户端的播放速率(即本地时钟频率)、传输延迟抖动和传输过程中数据单元的丢失情况不同，各用户的播放过程可能不同步。为了检测同步状况，用户端在播放某个 LDU 的同时，将该 LDU 的 RTS 反馈给服务器。反馈可以隔一段时间进行一次，且信息量很小，并不会显著加重网络的负担。服务器周期性地比较反馈回来的 RTS 值，就可以检测出各条流之间的同步情况。

设 f_a 和 f_b 为 A、B 两地返回的反馈单元，它们到达服务器的时刻分别为 τ_a 和 τ_b，它们最早和最晚发送时刻 t^e 和 t^l 则分别为

$$t^e(f_a)=\tau_a-d_{\max} \tag{6-15}$$

$$t^l(f_a)=\tau_a-d_{\min} \tag{6-16}$$

$$t^e(f_b)=\tau_b-d_{\max} \tag{6-17}$$

$$t^l(f_b)=\tau_b-d_{\min} \tag{6-18}$$

既然 f_a 和 f_b 的真正发送时刻 $t(f_a)$ 和 $t(f_b)$ 分别在(6-15)式与(6-16)式、(6-17)式与(6-18)式所定义的区间内，则有

$$t^l(f_b)-t^e(f_a)\geqslant t(f_b)-t(f_a) \qquad (\tau_b\geqslant\tau_a \text{ 时}) \tag{6-19}$$

$$t^l(f_a)-t^e(f_b)\geqslant t(f_a)-t(f_b) \qquad (\tau_b<\tau_a \text{ 时}) \tag{6-20}$$

由(6-19)式和(6-20)式可得[13]

$$\max\{t^l(f_b)-t^e(f_a),t^l(f_a)-t^e(f_b)\}\geqslant|t(f_a)-t(f_b)| \tag{6-21}$$

只要上式左侧满足

$$\max\{t^l(f_b)-t^e(f_a),t^l(f_a)-t^e(f_b)\}\leqslant\varepsilon \tag{6-22}$$

则

$$|t(f_b)-t(f_b)|\leqslant\varepsilon \tag{6-23}$$

$t(f_a)$ 和 $t(f_b)$ 反映了 A、B 两点的播放进度。如果 A、B 两地的播放已精确地同步，RTS 相同的反馈单元的发送时刻应该相等，即 $t(f_a)=t(f_b)$。服务器根据 RTS 相同的反馈单元的到达时刻 τ_a 和 τ_b 按 (6-15) 式 ～ (6-18) 式进行计算，并判断(6-22)式是否成立。式中 ε 表示流间同步所允许的最大偏差(同步容限)。如果(6-22)式不成立，则说明 A、B 两地的播放不同步，需要进行调整。

在考虑媒体流之间的时间约束关系时，通常选择其中一条流的时间轴作为基准，这条流称为**主**

流，其余流称为**从流**。当检测到媒体流之间的同步关系遭到破坏时，则保持主流的播放速率不变，调整从流的播放速率。比较从流与主流速率的快慢，然后加速或减缓从流播放速度，也可以暂停、重复或跳过某些从流数据单元，以达到与主流的一致。

主流的选择一般根据媒体流的重要性、时钟的精确程度等因素来确定。对于音频流和视频流而言，由于听觉对声音的不连续性比视觉对图像不连续的敏感程度要高，因而通常选择音频流为主流，视频流为从流。当然，如果具体应用中有某些特殊的调整功能，例如可以调整声音的静默期的长短的话(听觉对静默期长短的变化不敏感)，也可以做相反的选择。

6.6 接收与发送时钟的同步

6.6.1 基于接收缓存器的方法

在多媒体系统中，时钟是时间计量的基准，因此在讨论时钟同步问题时我们常常从时间、而不是从频率的角度来对两个时钟进行比较。在某个时刻 t，两个时钟显示的时间差别 $\Delta T(t)$ 称为二者的相对**时间偏差**(offset)，单位为秒，即

$$\Delta T(t) = C_1(t) - C_2(t) \tag{6-24}$$

式中，$C_1(t)$ 和 $C_2(t)$ 分别为 t 时刻时钟 1 和 2 显示的时间。由于时间是以时钟脉冲的周期来计量的，如果两个时钟的频率不同，则二者的时间变化快慢(速度)不同，这个差别 $\Delta T'(t)$ 称为相对**计时速度偏差**，单位是秒/秒，也可以称为相对**频率偏差**(skew)，即

$$\Delta T'(t) = C_1'(t) - C_2'(t) = \alpha C_2'(t) - C_2'(t) = (\alpha - 1)C_2'(t) \tag{6-25}$$

式中，α 是时钟 1 和时钟 2 之间的频率之比。两个时钟计时速度(频率)变化率的差别 $\Delta T''(t) = C''_1(t) - C''_2(t)$，称为二者的相对**频率漂移**(drift)。频率漂移与时钟的稳定性有关。

收、发时钟之间如果只存在固定的时间偏差不会对多媒体同步造成破坏。如果二者之间存在频率偏差，则产生图 6-19 所示的现象。两个时钟之间的频率漂移对多媒体同步的影响比较复杂，而且现代电子技术可以保证振荡器的频率漂移限制在较小的范围之内，因此本节将不考虑频率漂移，而只讨论收、发时钟之间存在固定频率偏差情况下的多媒体同步问题。

一个多媒体系统所能容忍的收发时钟频率偏差由具体应用而定。例如，假设发端实时地向收端发送数据，且收、发两端的时钟偏差为 10^{-3} 秒/秒，对播放一个 90 分钟的视频节目来讲，在节目的最后，两端时间之差为 5.4 秒，显然播放的质量难以得到保证。如果偏差为每秒 10^{-6} 秒，节目最后收、发端的时间之差为 5.4 ms，对媒体同步的影响就不易察觉出来了。下面我们来考虑收、发时钟频率偏差不能忽略的情况，即在图 6-19 所示的情况下，设计一个合适的起始时刻偏移量和缓存器容量，以避免出现数据尚未到达或缓存器溢出的问题。

假设媒体流的 LDU 按发送时钟的节拍进入接收缓存器 B，然后以接收时钟的节拍输出进行播放。在不考虑延时抖动的情况下，按发送时钟 C_s 计量的 LDU 到达时间间隔等于其发送的时间间隔

$$a_s(i) - a_s(1) = t_s(i) - t_s(1) \tag{6-26}$$

第 i 个 LDU 在缓存器中缓存的时间为 $[p_r(i) - a_s(i)]$，其中 $p_r(i)$ 为按接收时钟 C_r 计量的第 i 个 LDU 的播放时刻。为了平滑由收、发时钟频率偏差而引起的数据波动，所需的缓存器容量为

$$B = \max\{p_r(i) - a_s(i) - [p_r(1) - a_s(1)] \mid i \in (2,3,\cdots)\} \tag{6-27}$$

上式 $[\cdot]$ 中的项为起始时刻偏移量 D，$\{\cdot\}$ 中的项表示在播放开始之后，第 i 个 LDU 在缓存器中继续缓存的时间。如果收、发时钟是同步的，$p_r(i) - p_r(1) = a_s(i) - a_s(1) = t_s(i) - t_s(1)$，$B = 0$。这说明在接收端的播放速率和发送端的发送速率相同时，数据不会在缓存器中累积(或过快地消耗)，因此不必在起始偏移量 D 之外设置额外的缓存器。

假设发、收时钟之间的频率之比为 $\alpha = C'_s/C'_r$，那么用发送时钟 C_s 来度量时间间隔 $[p_r(i)-p_r(1)]$，我们有

$$p_r(i) - p_r(1) = \alpha[t_s(i) - t_s(1)] \tag{6-28}$$

将(6-26)式和(6-28)式代入(6-27)式，得到

$$\begin{aligned} B &= \max\{(\alpha-1)[t_s(i)-t_s(1)] \mid i \in (2,3,\cdots)\} \\ &= (\alpha-1)T_N \end{aligned} \tag{6-29}$$

式中，T_N 为媒体流的最长持续时间，$(\alpha-1)$反映了收、发时钟之间的频率偏差。当 $\alpha>1$ 时，接收时钟慢于发送时钟，上式给出防止缓存器溢出所需要的额外容量(总量为 $D+B$)。当 $\alpha<1$ 时，接收时钟快于发送时钟，此时 B 为负值，代表为了预防缓存器变空在开始播放(起始时刻 D)之前应该预先缓存的数据量。

当 α 偏离 1 较大时，补偿收、发时钟频率偏差所需要的缓存量可能过大，此时可以采用 6.4.2 节的方法来进行同步。当接收时钟慢于发送时钟时，缓存量总是趋于上警界线；反之，则总是趋于下警界线。

6.6.2 基于时间戳的锁相方法

在有些多媒体应用中，对收、发时钟的同步有着严格的要求。例如，中心站通过长途干线或卫星线路接收数字电视，然后转换成模拟电视信号送给模拟有线电视网络。由于模拟电视的同步信号和副载波信号由解码时钟产生，而副载波的相位和幅度代表彩色的色调和饱和度(见 2.2.5 节)，因此收、发时钟的频率偏差和漂移会造成重建图像的彩色失真或周期性变化(color cycling)、声音断续以及画面跳动等。

在需要收、发时钟频率精确同步的时候，锁相是一种通常采用的方法，它能够消除收、发时钟之间的频率偏差和频率漂移。图 6-26 所示是一个典型的锁相环路，它由相位比较器、环路(低通)滤波器和本地压控振荡器 VCO 三部分组成。锁相的基本过程是，将发送端时钟频率(由输入信号所携带)与接收端本地时钟频率在相位比较器中进行相位比较，两端时钟频率的偏差将反映为相位之差；相位比较器的输出参量(如电压大小、或脉冲宽窄等)，经环路滤波器滤除高频分量(输入相位的高频抖动)以后，控制 VCO 的振荡频率，使之与输入频率相等。锁相环路可以是模拟的，也可以是数字的。

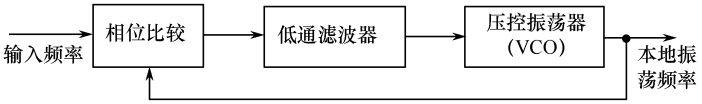


图 6-26　一般的锁相环路

当媒体流采用分组方式传输时，输入信号不是连续的正弦波或脉冲串，无法与本地时钟信号进行相位比较，此时需要有一种合适的方法来检测收、发时钟的频率偏差以控制 VCO 的振荡频率。在分组传输中，接收端可能获得发送时钟信息的途径有两个：(1)包的到达频率；(2)包头携带的表示该包发送时刻的时间戳。如果在接收端设一个输入缓存器，数据包按发送时钟速率到达缓存器，而以接收时钟速率被取出，那么缓存器的充满程度就反映了收、发时钟的频率差异。但是其他原因，如网络传输的延时抖动和包丢失等也会影响缓存器的充满程度，从而干扰锁相环的工作。比较好的获得发送时钟信息的途径是检测时间戳。例如 MPEG-2 标准中规定，编码器必须在不超过 0.1 秒的时间间隔内，在码流中传送一个节目时钟参考 PCR(见 8.2.2 节)。PCR 的数值等于 PCR 的最后一个字节产生时的时钟计数值，因此我们也可以将其看作为一种时间戳。基于时间戳的“锁相环路”如图 6-27 所示。时间戳检测模块检测所接收的数据包中的 PCR，然后计算 PCR 与本地时钟计数值之间的差值，如果收、发时钟频率没有偏差，该差值为一常数；如果有偏差，该差值经滤波

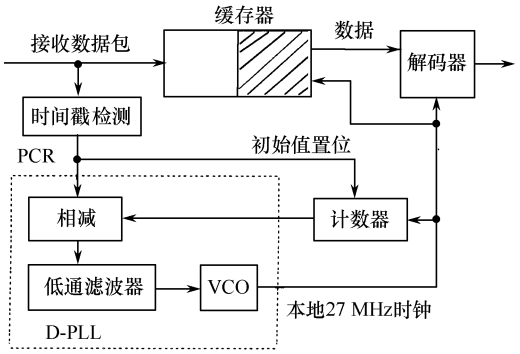


图 6-27　基于时间戳的时钟同步

后控制本地时钟频率，使本地时钟达到与发送端时钟频率的一致。

必须注意，在图 6-27 所示的电路中，用来控制本地时钟频率的计数差值实际上是包的发送时刻(PCR)和到达时刻(本地时钟计数)之间的差值，如果传输延时存在抖动的话，即使收、发时钟频率相同，该差值也不是恒定的。因此环路滤波器必须精心设计以便能够滤除传输延时抖动的影响[14]，否则延时抖动将通过环路转化为本地时钟频率的晃动。

6.6.3　基于网络时间协议的方法

在因特网中，常常通过**网络时间协议** NTP(Network Time Protocol)[15]来解决一对站点、或多个站点间的时钟频率偏差问题。具体做法是，由中央时间服务器维护一个高精确度和高稳定度的时钟(网络时钟)，各站点将此信号作为调整本地时钟的基准，图 6-28 说明获得时间基准的过程。客户端在本地时间 A 向 NTP 服务器发送一个带有发送时刻 A 的 NTP 包，服务器在本地时间 B 收到该包，然后在本地时间 C 将一个带有时刻 A、B 和 C 的包发送回客户端，客户端在本地时间 D 收到。假设客户端到服务器的实际单程延时为 d，在这段路程上由收、发时钟频率不同而产生的时间偏差为 offset，则 $B-A=d+\text{offset}$，且 $D-C=d-\text{offset}$，

$$d=(B+D-A-C)/2 \tag{6-30}$$

$$\text{offset}=(B+C-A-D)/2 \tag{6-31}$$

(6-31)式检测到的时间偏差相当于一般锁相环中相位比较器的输出。

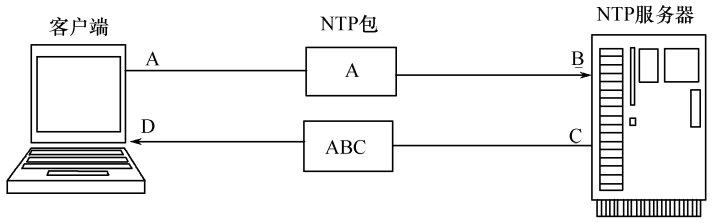


图 6-28　NTP 包的传送

在实际应用中，检测到的延时和时间偏差会受到延时抖动的影响。NTP 规定了一个“锁相环路”的结构[15]，在这个结构中，利用滤波、选择和加权等方法得到最为可靠的时间偏差值，然后该值经环路滤波器控制本地的 VCO。经过这样的调整，各站点的时钟同步的精度可保持在 10 ms 之内。

6.7　同步算法小结

对于流内同步而言，同步机制的目标是保持 LDU 之间均匀的时间间隔，而接收端的缓存器是滤除不均匀性、保证流内同步的重要工具。在播放器按均匀时间间隔从缓存器取走 LDU 的同时，由于收发时钟频率的不同、网络传输延时的抖动和丢包等原因，LDU 进入缓存器的时间间隔与播放时间间隔不一致，因此缓存器充满程度的变化常被作为检测同步状况的标志，防止缓冲器的上溢和下溢则是同步机制设计中需要考虑的主要问题。

对于流间同步，我们关心的是不同流中相对应的 LDU(如视频的一帧和该帧对应的那段伴音)

应在相同的时刻播放。为此，一个广泛采用的方法是，为各个流的 LDU 打上相对时间戳，并选择其中一个流为主流，其余的为从流，流间同步机制则以主流的时间轴为基准，保证所有流中具有相同时间戳的 LDU 同时播放。

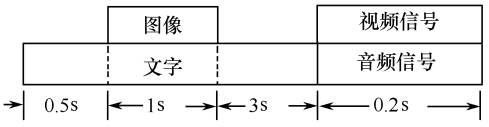
调整收、发时钟频率，或增/删缓存器中的数据单元是进行同步（无论是流内，还是流间）调整的一个常用手段。此时需要注意，人们对音频的不连续性比对视频的敏感，而对音频频率变化的容忍程度又比对波形变化的要低。一般来说，增/删数据比改变频率容易实现，且对主观听、视觉的影响较小。

当系统中有多个信源和/或多个信宿时，组同步需要让各个组成员有一个共同的起始参考时间，这通常通过互相交换一些协议信息（称为启动协议）来实现。6.5.1 节中的启动器就是一个这样的例子。组同步的另一要求是组内各信宿的同步播放。

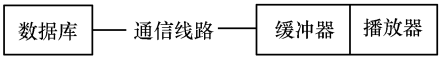
在同步机制中常常使用反馈环路。任何一个反馈环路（大到将传输线路包括在内，小到集成电路中的锁相环）应该包括**状态检测**（如图 6-22 中的缓存器、图 6-26 中的相位比较器）、**环路滤波器**和**状态调整**（如图 6-22 的控制函数、图 6-26 中的压控振荡器）三部分。环路滤波器是环路设计中最重要的部分，它影响环路对状态变化响应的灵敏程度和环路的稳定性。

习 题 六

1. 多媒体数据内部有哪些约束关系？这些约束关系分别是怎样形成的？它如何体现了 1.2 节中所讲的“多媒体数据是由内容上相关联的多种媒体数据所形成的复合数据，这一数据合成过程是在计算机控制下完成的”的特点？
2. 多媒体数据的主要构成部分是哪些？多媒体同步主要研究哪些问题？
3. 定义时域参考框架的意义是什么？它包括哪些部分？相互有什么关系？
4. 试用时间轴模型和 OCPN 两种模型描述习题 4 图中的时域场景，其中视频和音频数据流之间的同步偏差要求小于等于40 ms（这需要 40 ms 检查一下同步情况）。
5. 一个远端数据库内存有一个 60s 的、频率为 30 帧/s 的视频短节目。该节目以 MPEG-1 压缩方式存储，每帧平均数据量为 50 kb。此节目经由一条带宽为 512 kb/s 的通信线路传送到接收端（见习题 5 图）。



习题 4 图

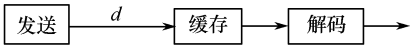


习题 5 图

（1）在数据库内此节目所需的存储空间（MB）是多大？

（2）此线路能否实时地传输节目？如果不能，应如何解决这个问题？计算接收端的起始延迟时间以及为保证接收端正常播放所需要的缓存器的大小。

6. 在习题 6 图所示的具有全局时钟的实时多媒体系统中，发送端将压缩后的每一帧图像打在一个包中传送，每个包的包头有表示该包发送时间的时间戳 $T(i)$，$i=1,2,3\cdots$



习题 6 图

（1）保证接收端连续播放的条件是什么？

（2）若网络传输延时 $d(i)\in[\Delta_{\min},\Delta_{\max}]$，为保证播放的连续性，接收端应该在什么时刻开始播放？其缓存器应为多大？与公式(6-8)和公式(6-9)表示的结果有什么相同和不同？

（3）假设网络传输延时 d 为一符合正态分布的随机变量，其均值为 μ，方差为 σ^2，在保证连续播放概率 $\eta\geqslant 99\%$ 的条件下，接收端应该在什么时刻开始播放（用解析式表达）？

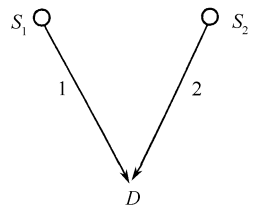
7. 比较基于播放时限和基于缓存数据量的两种流内同步方法。它们各适用于哪些情况？
8. 图 6-28(a)的系统中，为什么必须要有启动器？在图 6-29 所示的系统中是否需要启动协议？说明原因。

9. 有如下图所示的多媒体信息查询系统，用户所需的信息由视频流和音频流按一定的时间关系复合而成，但这两类信息却分别存储在不同的地点，试问：

(1)影响收端进行同步播放的因素有哪些？

(2)假设收、发端时钟频率无偏差，支路 1、2 的延时范围分别为 $[\Delta_{min},\Delta_{max}]$ 和 $[\delta_{min},\delta_{max}]$，且 $\delta_{max}>\Delta_{max}$，$\delta_{min}<\Delta_{min}$，同时假设每个 LDU 包头中有记录发送时刻 $t(i)$ 的时间戳，请给出保证视频和音频数据流在接收端能同步播放的起始延时时间。

(3)如果在播放过程中，发现音频流超前于视频流，你建议如何调整？



习题 9 图

10. 服务器按自己的时钟向客户端传输一个长度为半小时、帧率为 25 帧/秒、码率为 128 kb/s 的视频节目，客户端按本地时钟播放，二者的时钟偏差为 10^{-4} 秒/秒(服务器慢于客户端的)。假设客户端收到第一帧数据就开始播放，能否在保证观看质量的情况下连续播放完整个节目？假设累积时间差小于一个帧周期不会明显影响播放的观看质量，试提出一种起始延时最小且实现起来最简单的能保证客户端连续播放质量的方法(不考虑网络延时抖动)。

参 考 文 献

[1] G. Blakowski and R. Steinmetz, "A media synchronization survey: reference model, specification and case studies," IEEE JSAC, 1996, Vol. 14, No. 1, 5-35

[2] W. Appelt, Hyper ODA, ISO/IEC/JTC1/SC18/WG3, 1996

[3] M. J. Perez-Luque and T. D. C. Little, "A temporal reference framework for multimedia synchronization," IEEE JSAC, 1996, Vol. 14, No. 1, 36-51

[4] C. L. Hamblim, "Instants and intervals," Proc, 1st Conf. Int. Soc. Study of Time, New York, 1972, 324-331

[5] P. V. Beek and R. Cohen, "Exact and approximate reasoning about temporal relations," Comp. Intel., 1990, No. 6, 132-144

[6] S. Gibbs et al., "An object oriented framework for multimedia composition and synchronization," Proc. Eur. Multimedia. Workshop, Stockholm, Sweden, 1991

[7] ISO/IEC, Hypermedia/Time－Based Document Structuring Language (HyTime), International Standard ISO/IEC IS 10744, 1992

[8] T. D. C. Little and A. Ghafoor, "Synchronization and storage models for multimedia objects," IEEE J, SAC, 1990, Vol. 8, No. 3, 413-427

[9] H. Santoso, et al., "Preserving temporary signature: A way to convey time-constrained flows," IEEE Globecom, Houton Texas, 1993, 872-876

[10] E. Biersack and W. Geyer, "Synchronized delivery and playout of distributed stored multimedia streams," Multimedia Sys. 1999, Vol. 7, 70-90

[11] K. Rothermel and T. Helbig, "An adaptive protocol for synchronizing media streams," Multimedia Sys. 1997, Vol. 5, 324-336

[12] J. Escobar et al., "Flow synchronization protocol," IEEE/ACM Trans. Networking, 1994, Vol. 2, No. 2, 111-121

[13] P. V. Rangan. et al., "Techniques for multimedia synchronization in network file systems," Comp. Comm., 1993, Vol. 16, No. 3, 168-176

[14] G. F. Andreotti, etal., "Clock recovery and reconstruction of PAL pictures for MPEG coded strearns transported over ATM networks," IEEE Trans. CSVT, Vol. 5, No. b, 1995, 508-514

[15] D. L. Mills, "Internet time synchronization: The network time protocol," IEEE Trans, Comm. 1991, Vol. 38, No. 10, 1482-1493

第 7 章　多媒体传输网络

7.1　概述

要实现分布式的多媒体应用，例如 IPTV、多媒体会议等，必须用网络将处于不同地理位置的多媒体终端、服务器等设备连接起来，并使这些设备相互之间能够进行所需要的多媒体信息的传输。一方面由于多媒体数据的集成性，使它的传输既不像传统的通信业务那样，在一次呼叫中只传送一种媒体，例如只传声音、或是文字（传真），也不像计算机通信那样单纯传送数据，多媒体通信在一次呼叫中需要传送由多种媒体复合构成的信息。另一方面，在传统有线电视网上传输的虽然是声音和图像两种媒体，但这种网络是单向的，不支持多媒体的交互功能 。因此，利用传统的网络（无论是通信网，还是计算机网、电视广播网）进行多媒体信息传输，都不是理想的解决方案；但是从社会和经济的角度来看，多媒体信息的传输又不能完全摆脱这些具有长期历史的、已经“无处不在”的传统网络。这就形成了这一问题的复杂性。将这些针对不同应用目标设计的网络“融合”在一起以形成理想的多媒体业务网，虽然近年来已经取得了长足的进展，但要达到最终的目的还需要相当的过程。

在这里，我们所说的支持多媒体信息传输的“网络”是指这样几个部分：

（1）连接多媒体终端和网络节点之间以及节点与节点之间的传输介质和光、电部件，如光缆、电缆、无线信道、中继器、收/发设备等；

（2）在网络节点上接收一条链路上传来的信息、并将其转发到另一条链路上的交换设备，如各种类型的交换机、路由器、基站等；

（3）保障终端之间能够进行信息传输与交换的通信协议；

（4）网络服务与管理系统。

本章将从多媒体信息传输的角度对现有的主要网络及其演进过程进行讨论。

7.2　多媒体信息传输对网络的要求

7.2.1　性能指标

1. 吞吐量

吞吐量是指网络传送二进制信息的速率，也称比特率，或带宽。带宽从严格意义上讲是指一段频带，是对应于模拟信号而言的，在一段频带上所能传送的数据率的上限由香农信道容量所确定。不过通常在讨论数据传输时也常简单地说带宽，即指比特率。有的多媒体应用所产生的数据速率是恒定的，称为**恒比特率**（CBR）应用；而有的应用则是**变比特率**（VBR）的。衡量比特率变化的量称为**突发度**（Burstness）：

$$\text{突发度} = \frac{\text{PBR}}{\text{MBR}} \tag{7-1}$$

式中，MBR 为整个会话（Session）期间的平均数据率，而 PBR 是在预先定义的某个暂短时间间隔内的峰值数据率。支持不同应用的网络应该满足它们在吞吐量上的不同要求。

持续的、大数据量的传输是多媒体信息传输的一个特点。从单个媒体而言，实时传输的活动图像是对网络吞吐量要求最高的媒体。更具体一些，按照图像的质量我们可以将活动图像分为 5 个级别：

(1) 高清晰度电视(HDTV)。例如，分辨率为 1 920×1 080，帧率为 60 帧/秒，当每个像素以 24 比特量化时，总数据率在 2 Gb/s 的数量级。如果采用 MPEG-2 压缩，其数据率大约在 20～40 Mb/s；

(2) 演播室质量的普通电视。其分辨率采用 CCIR 601 格式。对于 PAL 制式，在正程期间的像素数为 720×576，帧率为 25 帧/秒(隔行扫描)，每个像素以 16 比特量化，则总数据率为 166 Mb/s。经 MPEG-2 压缩之后，数据率可达 6～8 Mb/s；

(3) 广播质量的电视。它相当于从模拟电视广播接收机所显示出的图像质量。从原理上讲，它应该与演播室质量的电视没有区别，但是由于种种原因(例如接收机分辨率的限制)，在接收机上显示的图像质量要差一些。它对应于数据率在 2～6 Mb/s 左右的经 MPEG-2 压缩的码流；

(4) 录像质量的电视。它在垂直和水平方向上的分辨率是广播质量电视的二分之一，经 MPEG-1 压缩之后，数据率约为 1.4 Mb/s(其中伴音为 200 kb/s 左右)；

(5) 会议质量的电视。会议电视可以采用不同的分辨率，我们这里指的是 CIF 格式，即 352×288 的分辨率，帧率为 10 帧/秒以上，经 H.261 标准的压缩后，数据率为 128～384 kb/s(其中包括声音)。在手机等低端设备中进行可视电话或会议时，可采用 QCIF 格式，经 H.263 或 MPEG-4 压缩后，数据率为 64 kb/s 左右。

声音是另一种对吞吐量要求较高的媒体，它可以分为如下 4 个级别：

(1) 话音。其带宽限制在 3.4 kHz 之内，以 8 kHz 取样、8 比特量化后，有 64 kb/s 的数据率。经压缩后，数据率可降至 32 kb/s、16 kb/s，甚至更低，如 4 kb/s；

(2) 高质量话音。相当于调频广播的质量，其带宽限制在 7 kHz，以 22kHz 取样、16 比特量化，数据率为 352 kb/s。经压缩后，数据率为 48～64 kb/s；

(3) CD 质量的音乐。它是双声道的立体声，带宽限制为 20 kHz，经 44.1 kHz 取样、16 比特量化后，每个声道的数据率为 705.6 kb/s。在使用 MUSICAM(见 5.9 节)压缩之后，两个声道的总数据率可降低到 192 kb/s。更高层次的音频压缩方法还可将其速率降到 128 kb/s，音乐质量仍可接近于 CD；而要得到演播室质量的声音时，数据率则为 CD 质量声音的 2 倍；

(4) 5.1 声道立体环绕声道的带宽为 3～20 kHz，取样率为 48 kHz，每个样值量化到 22 比特，采用 AC-3(见 5.11 节)压缩后，总数据率为 320 kb/s。

图 7-1 综合表示出不同媒体对网络吞吐量的要求，其中高分辨率文档是指分辨率在 4096×4096以上的图像(例如某些医学图像)。图中 CD 音乐和各种电视信号都是指经过压缩之后的数据率。由图看出，文字浏览对传输速率的要求是很低的。

2. 传输延时

网络的**传输延时**(Transmission Delay)定义为信源发送出第一个比特到信宿接收到第一个比特之间的时间差，它包括电(或光)信号在物理介质中的**传播延时**(Propagation Delay)和数据在网中的处理延时(如复用/解复用时间、在节点中的排队和切换时间等)。

另一个经常用到的参数是**端到端的延时**。它通常指一组数据在信源终端上准备好发送的时刻，到信宿终端接收到这组数据的时刻之间的时间差。端到端的延时，包括在发端数据准备好而等待网络接受这组数据的时间(Access Delay)、传送这组数据(从第一个比特到最后一个比特)的时间和网络的传输延时三部分。在考虑到人的视觉、听觉主观效果时，端到端的延时还往往包括数据在收、发两个终端设备中的处理时间，例如，发、收终端的缓存器延时、音频和视频信号的压缩编码/解码时间、打包和拆包延时等。

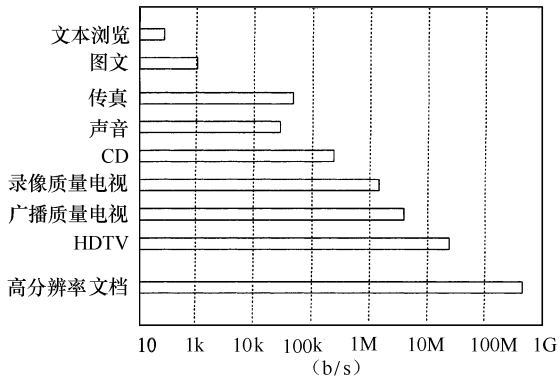


图 7-1　不同媒体对带宽的要求

对于实时的会话应用，ITU-T 规定，当网络的单程传输延时大于 24 ms 时，应该采取措施（使用方向性强的麦克风和喇叭、或设置回声抑制电路）消除可听见的回声干扰。在有回声抑制设备的情况下，从人们进行对话时自然应答的时间考虑，网络的单程传输延时允许在100～500 ms 之间，一般应小于 250 ms。在查询等交互式的多媒体应用中，系统对用户指令的响应时间也不应太长，一般应小于 1～2 秒。如果终端是存储设备或记录设备，对传输延时就没有严格要求了。

3. 延时抖动

网络传输延时的变化称为网络的**延时抖动**。度量延时抖动的方法有多种，其中一种是用在一段时间内（如一次会话过程中）最长和最短的传输延时之差来表示。

产生延时抖动的原因可能有如下一些：

（1）传输系统引起的延时抖动，例如符号间的相互干扰、振荡器的相位噪声、金属导体中传播延时随温度的变化等。这些因素所引起的抖动称为**物理抖动**，其幅度一般只在微秒量级，甚至于更小。例如，在本地范围之内，ATM 交换机工作在 155.52 Mb/s 时，最大的物理延时抖动只有 6 ns 左右（不超过传输 1 个比特的时间）。

（2）对于电路交换的网络（如公用电话网），只存在物理抖动。在本地网之内，抖动在毫微秒量级；对于远距离跨越多个传输网络的链路，抖动在微秒的量级。

（3）对于共享传输介质的局域网（如以太网），延时抖动主要来源于**介质访问时间**（Medium Access Time）的变化。终端准备好欲发送的信息之后，必须等到共享的传输介质空闲时，才能真正进行信息的发送，这段等待时间就称为介质访问时间。

（4）对于广域的分组网络（如 IP 网），延时抖动的主要来源是流量控制的等待时间（终端等待网络准备好接收数据的时间）的变化和存储转发机制中由于节点拥塞而产生的排队延时的变化。在有些情况中，后者可长达秒的数量级。

在第 6 章中我们已经知道，延时抖动将破坏多媒体的同步，从而影响音频和视频信号的播放质量。例如，声音样值间隔的变化会使声音产生断续或变调的感觉；图像各帧显示时间的不同也会使人感到图像停顿或跳动。人耳对声音的变化比较敏感，如果从你所熟悉的音乐中删节很小的一段，例如 40 ms，便立刻会有所感觉。人眼对图像的变化则没有这样敏感，在熟悉的录像片中间删节掉 1 秒钟（无伴音时）长的一段，你未必感觉出来[1]。因此，声音的实时传输对延时抖动的要求比较苛刻。尽管可以用一定的方法在终端对网络的延时抖动给予补偿（见第 6 章），但补偿需要使用大的缓存器，因而会增加端到端的延时。在考虑到实际应用对缓存器大小和延时时间所能承受的限制下，下述定量指标（补偿前的数值）可以作为参考[1]：对于经压缩的 CD 质量的声音，网络延时抖动一般应不超过 100 ms；对于电话质量的语声，抖动不应超过 400 ms；对于对传输延时有严格要求的

应用，如虚拟现实，抖动不超过 20～30 ms。由于活动图像总是和伴音一起传送的，我们可以从对伴音的要求推导出对视频信号传输的延时抖动的要求（补偿前的数值）：对于已压缩的 HDTV，网络延时抖动应不超 50 ms；对于已压缩的广播质量的电视，不超过 100 ms；对于会议电视，不超过 400 ms。

对于文字、图形、图像等静态媒体的传输，网络的延时抖动不产生什么影响。

4. 错误率

在传输系统中产生的错误由以下几种方式度量：

（1）**误码率** BER(Bit Error Rate)，指在从一点到另一点的传输过程（包括网络内部可能有的纠错处理）中所残留的错误比特的频数。BRE 通常主要衡量的是传输介质的质量。对于光缆传输系统，BER 通常在 10^{-12}～10^{-9} 之间；而在无线信道上，BER 可能达到 10^{-4}～10^{-3}，甚至 10^{-2}。

（2）**包错误率** PER(Packet Error Rate)，是指同一个包两次接收、包丢失、或包的次序颠倒而引起的包错误。包丢失的原因可能是由于包头信息的错误而未被接收，但更主要的原因往往是由于网络拥塞，造成包的传输延时过长、超过了应该到达的播放时限而被接收端舍弃，或网络节点来不及处理而被节点丢弃（节点缓存器溢出）。

（3）**包丢失率** PLR(Packet Loss Rate)，它与 PER 类似，但只关心包的丢失情况。

在多媒体应用中，将未压缩的声像信号直接播放给人看时，由于显示的活动图像和播放的声音是在不断更新的，错误很快被覆盖，因而人可以在一定程度上容忍错误的发生。从另一方面看，已压缩的数据中存在误码对播放质量的破坏显然比未压缩的数据中的误码要大，特别是发生在关键地方（如头信息运动矢量等）的误码要影响到前、后一段时间和/或空间范围内的数据的正确性。此外，误码对人的主观接收质量的影响程度还与压缩算法和压缩倍数有关。下面我们给出在一般情况下（即使用第 4 章讨论的典型算法和码率时）获得“好”的质量所要求的误码率指标[1]。对于电话质量的语声，BER 一般要求低于 10^{-2}。对未压缩的 CD 质量的音乐，BER 应低于 10^{-3}；对已压缩的 CD 音乐，应低于 10^{-4}。对于已压缩的会议电视，BER 应低于 10^{-8}；对已压缩的广播质量的电视，应低于 10^{-9}；对已压缩的 HDTV，则应低于 10^{-10}。如果对已压缩的视频码流采用**前向纠错**技术（见 9.3.2 节），可允许的误码率则大约为上述数据乘以 10^{4}。

与声音和活动图像的传输不同，数据对误码率的要求很高，例如银行转账、股市行情、科学数据和控制指令等的传输都不容许有任何差错。虽然物理的传输系统不可能绝对不出差错，但是可以通过检错、纠错机制，例如利用所谓**自动重发请求** ARQ（见 9.3.1 节）协议在检测到差错、包次序颠倒或超过规定时间限制仍未收到数据时，向发端请求进行数据重传，使错误率降为零。

7.2.2 网络功能

1. 单向网络和双向网络

单向网络指信息传输只能沿一个方向进行的网络。例如，传统的有线电视（CATV）网，信息只能从电视中心向用户传输，而不能反之。支持在两个终端之间、或终端与服务器之间互相传送信息的网络称为双向网络。当两个方向的通信信道的带宽相等时，称为**双向对称信道**；而带宽不同时，则称为**双向不对称信道**。由于多媒体应用的交互性，多媒体传输网络必须是双向的。

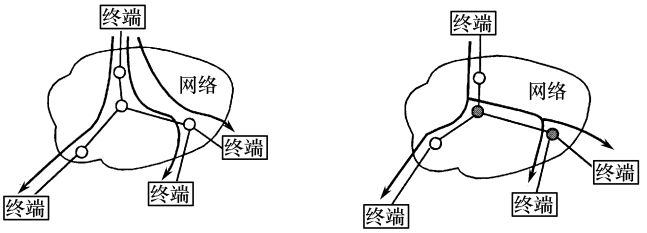
上述概念是从信道的角度来定义的。在有关通信的书籍中，还常常遇到单工、半双工和全双工的概念，这是从传输方法的角度来定义的。**单工**是信号向一个方向传输的方法；**半双工**是信号双向传输的方法，但在某一时刻只会朝一个方向传输；**全双工**是同时双向传输的方法。支持半双工传输的网络，例如传统的以太网，我们也认为它是双向网络。

2. 单播、多播和广播

单播（Unicast）是指点到点之间的通信；**广播**（Broadcast）是指网上一点向网上所有其他点传送

信息；**多播**(Multicast)、或称为组播，则是指网上一点对网上多个指定点(同一个工作组内的成员)传送信息。

发送终端通过分别与每一个组内成员建立点到点的通信联系，能够达到多播的目的。但是在这种情况下，发送端需要将同一信息分别送到多个信道上(见图 7-2(a))。同一信息的多个复制版本在网上传输，无疑要加重网络的负担。多播是指网络具备这样的能力：其中间节点能够按照发端的要求将欲传送的信息在适当的节点复制，并送给指定的组内成员，这也称为**多点路由功能**。图 7-2(b)给出了一个多播的例子，图中灰色圆点代表网络中进行信息复制的节点，粗箭头表示多播的数据流走向。



(a) 多个点到点的信道　　(b) 多播信道

图 7-2　多播

不同的多媒体信息系统需要不同的网络结构来支持。简单的可视电话只需要点对点的连接，而且这一连接是双向对称的。在多媒体信息检索或 VOD 系统中，用户和中心服务器之间建立的可能是点对点的联系，也可能是点对多点的联系(服务器向多个用户传送共同感兴趣的同一信息或节目)，但使用的信道都是双向不对称的。通常从用户到中心(**上行**)的线路只传送查询命令，所需要的带宽较窄；而从中心到用户(**下行**)传送大量的多媒体数据，需要占用频带较宽的线路。分配型的多媒体业务，例如数字电视广播，则需要广播型的网络。多点与一点连接的结构在有些情况下也会遇到，例如在信息检索系统中，如果数据库是分布式的，往往需要从多个库中调取信息来回答一个用户的要求。多媒体合作工作是对通信机制要求最高的应用，它要求多点对多点之间的双向对称连接。此时，多播功能是必须的。因此，支持综合多媒体业务的传输网络应当支持单播、多播和广播。

7.2.3　服务质量(QoS)和用户体验质量(QoE)的保障

1. QoS 定义

服务质量 QoS(Quality of Service)是多媒体网络中的一个重要的概念。在传统的网络中，没有提出 QoS 的概念，这是因为传统的网络都是针对某个专项业务的要求而设计的，例如，电话由公用电话网传送，一路电话占有固定的带宽(或时隙)，当终端进行呼叫时，网络不需要再询问终端对带宽的要求；而且一旦网络将此呼叫接通，在整个会话过程中，终端一直拥有这一带宽，服务质量也就自然是恒定地得到保障的。但是在一个支持综合业务的网上，由于要在同一个网络上支持不同的业务，而不同的业务对网络的性能又有不同的要求，因此有必要在开始实际的信息传输之前，将某项业务的特定要求告知网络，QoS 的概念也就随之出现。在多媒传输网络中，QoS 的概念则更进一步地受到重视。

ITU 将 QoS 定义为决定用户对服务的满意程度的一组服务性能参数。这些参数可以用多种方式来描述，例如确定性方法和统计描述方法。

确定性描述方法：
$$\begin{aligned} &\text{QoS_Parameter} \leqslant \text{Upper_bound} \\ &\text{QoS_Parameter} \geqslant \text{Lower_bound} \end{aligned} \tag{7-2}$$

统计描述方法：　Prob[QoS_Parameter ≤ Upper_bound]≥Prob_bound

Prob[QoS_Parameter≥Lower_bound] ≥ Prob_bound　　(7-3)

7.2.1 节中介绍的吞吐量、传输延时、延时抖动和错误率则是常用的网络 QoS 参数。

2. 定量 QoS 服务和定性 QoS 服务

从提供的**服务种类**来区分，QoS 服务可以分为定量和定性的两种。所谓定量服务，是指用户用确定性描述方法，或统计描述方法提出具体的 QoS 指标，网络在数据传输过程中保证满足这些指标。在定性服务中，对 QoS 参数不指定具体的指标，但是网络能够对某些应用提供比另一些应用更“好”的服务。一个典型例子就是网络提供不同优先级的服务，高优先级的数据优先得到服务而具有较小的延时，或较少的丢包率。

3. 针对流和针对包类型的 QoS 服务

从**服务对象**来区分，QoS 服务可以分为针对流(Per-flow)和针对包类型(Per-class)的两种。前者是网络对某一个应用产生的数据流提供同样(定量或定性)的 QoS 服务。显然，能够提供针对流的服务的网络(网络节点)必须首先能够识别和区分不同应用所产生的流。在针对包类型的服务中，所有的用户数据都根据某种准则，如应用类型、QoS 要求或协议簇等，被划分为几种类型，网络对相同类型的数据包(无论是哪个应用产生的)提供同样的服务。显然，这样的网络(网络节点)首先必须具备区分包类型的能力。针对包类型的服务通常是定性的 QoS 服务，为了与能够定量地保障带宽或延时的 QoS 服务相区别，也常称为 **CoS**(Class of service)服务。

4. QoS 机制

不同的多媒体应用对网络的性能有不同的要求。在通信起始时，用户向网络提交的 QoS 参数实际上描述了应用对网络资源的需求，网络可以以此作为对网络内部共享资源(如带宽、处理能力、缓存空间等)进行管理的依据。如果网络资源不能满足用户的 QoS 要求，或者接纳一个新的用户要侵犯预留给正在进行通信的用户的资源、从而降低这些通信的 QoS 的话，网络将不接纳这个新用户。这种机制通常称为**连接接纳控制**(Connection Admission Control，CAC)一旦网络接纳了某个用户的呼叫，它就有责任在整个会话过程中保障该用户所提出的 QoS 要求。因此，网络要为这个呼叫预留资源，并在通信过程中进行性能监控和资源的动态调整；当资源不能保障用户的 QoS 要求时，通知有关的用户，直至中止相关的通信。上述各种功能构成了网络的 **QoS 保障机制**。目前只有一些网络实现、或部分地实现了这些功能，QoS 保障的问题尚处于研究和发展之中。

5. QoE 概念

用户体验质量(Quality of Experience，QoE)是近年提出的一个新概念，用以度量用户获得和使用某项多媒体业务的主观满意程度。前述的网络 QoS 保障是 QoE 保障的一个重要组成部分，但二者并不等同。QoE 涉及从信息内容的制作到显示的整个端到端的业务键，它不仅限于对网络传输质量的要求，还包括用户对重建媒体质量的主观评价、对业务的性价比、使用的方便性和舒适度等方面的要求。因此，QoE 的保障机制不但需要考虑 QoS 机制、网络覆盖和终端性能等技术因素，还需要考虑业务的便利和快捷性、价格、客服支撑等非技术因素。

7.3　网络类别

在本节中将从不同的角度对现有的网络进行归类，以便对不同网络对多媒体信息传输的支持情况有一个总括的了解。

7.3.1 电路交换网络和分组交换网络

根据数据交换的方式，可以将现有的网络分成电路交换和分组交换两大类型。所谓**交换**是指在网络中给数据正确地提供从信源到信宿的路由，并引导数据通过此路由的过程。

在**电路交换**的网络中，一旦两个终端之间通过信令协议建立起了一条经过网络的通信连接，它们之间就独占了一条物理信道[见图 7-3(a)]。在频分复用(FDM)的信道中，这个“物理信道”意味着一个固定的频带；而在时分复用(TDM)系统中，则意味着一个固定的时隙。即使这一对用户进行信息交换的速率低于信道提供的速率，甚至停止信息交换(如说话的间隙)、信道处于空闲状态，只要用户不通知网络撤销这个连接，该信道就不能为其他用户所使用。同时，这一对用户开始通信后，不管网络变得多么繁忙，该用户所独占的资源(传输速率)也不会被其他后来的用户所侵占。当网络不能给更多的用户提供信道时，则只能简单地不接纳后来的呼叫。

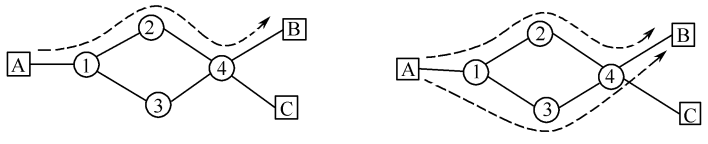
从多媒体信息传输的角度考虑，电路交换网络的优点是：① 在整个会话过程中，网络所提供的固定的比特率是得到保障的；② 路由固定，虽然信号传输之前存在一段建立通信连接的延时，但连接建立后传输延时短，延时抖动只限于物理抖动。这些都有利于固定比特率的连续媒体的实时传输。电路交换网络的缺点是：①带宽利用率较低；②不支持多播，因为这些网络原来是为点到点的通信而设计的。当多媒体应用需要多播功能时，必须在网络中插入特定的设备，称为**多点控制单元** MCU(Multipoint Control Unit)，在 7.4.2 节中，将对 MCU 作进一步讨论。

普通公用电话网(PSTN)、窄带综合业务网(N-ISDN)、数字数据网(DDN)和第 1 和 2 代的蜂窝移动网等都属于电路交换网络。在带宽不成问题的新出现的光网络中，也使用电路交换。

分组交换也称为包交换。在分组交换网络中，信息不是以连续的比特流的方式来传输的，而是将数据流分割成小段，每一段数据加上头和尾，构成一个包、或称为分组(在有的网络中称为帧、或信元)，一次传送一个包。当长距离传输需要经过网络的中间节点时，节点先将整个包存储下来，然后再转发到适当的路径上，直至到达信宿，这通常称为**存储－转发机制**。分组交换网络的一个重要特点是，多个信源可以将各自的数据包送进同一线路，当其中一个信源停止发送时，该线路的空闲资源(带宽)可以被其他信源所占用，也就是说，其他信源可以传送更多的数据，这就提高了网络资源的使用效率。但是这种复用一般来说是统计性的。在某个信源的通信过程中，如果有过多的其他信源加入网络，则该信源的资源可能被其他信源所侵占，导致它所要求的比特率得不到保障，传输延时加长；如果在某个时刻多个信源同时送进过多的数据，还可能造成网络负荷超载的情况。

根据节点对包的处理方式不同，分组交换可以分成两种工作模式：数据报(Datagram)和虚电路(Virtual Circuit)。在**数据报**模式中，为了将同一线路上的不同信源的数据包区分开，每一个数据包的包头中都含有信宿的标识，网络根据标识将数据包正确地送至目的地，这同邮局按照用户写在信封上的地址送信的过程类似。由于节点为每个包独立地寻找路径，因此送往同一信宿的包可能通过不同的路径传到信宿[见图 7-3(b)]。在**虚电路**模式中，两个终端在通信之前必须通过网络建立逻辑上的连接；连接建立后，信源发送的所有数据包均通过该路径顺序地传送到信宿(无需再对每个包作路由选择)；通信完成后拆除连接。这与电路交换的方式很相似，但其根本的区别是，节点对包的处理采用的仍是存储－转发机制。这个逻辑上的连接称为虚电路，在本书中也称为逻辑信道。值得指出，在通信书籍中讨论网络的交换问题时，逻辑信道这个词与虚电路有不同的含义，它代表信道的一个编号，只有占用和空闲状态，不存在建立和拆除。一条虚电路则由各段的逻辑信道(号)连接而成。在本书中由于不讨论网络层的细节，所以常将这两个词混用。

分组交换网络的优点是：①带宽利用率高；②能够使用优先级别，网络节点可以首先传输优先级高的分组，从而提供定性的 QoS 服务；③有的分组交换网允许在一次连接中建立多条逻辑通道，这对多媒体信息的传输很有利。正如我们在 7.2.1 节中所看到的，实时媒体和静态媒体对网络性



A到B之间建立物理通道　　数据报模式

（a）电路交换网　　（b）分组交换网

图 7-3　电路交换与分组交换

能的要求有很大的差异，用同一个通道传送，则该通道的每一项指标都必须满足各成分数据中要求最高的那一种，才能保证各种媒体的良好传输。如果采用不同的逻辑通道分别传送具有不同 QoS 要求的媒体数据，则网络资源可以得到更合理的利用。分组交换网对多媒体信息传输不利之处是：①由于采用存储转发机制，传输延时较大，且存在抖动；②网络性能，即占有的比特率、传输延时和延时抖动随网络负荷的增加而恶化；严重时还产生包的丢失；③在数据报模式下，包的到达顺序可能不是它们的发送顺序。

局域网、传统的数据传输网络 X.25、帧中继，以及 IP 网都属于分组交换的网络。

7.3.2　面向连接方式和无连接方式

电路交换和分组交换中讨论的是信息在网络内部如何传送的，现在要讨论的则是连接问题，即在什么条件下网络才接受数据。

在**面向连接**的网络中，两个终端之间必须首先建立起网络连接，即网络接纳了呼叫并给予连接，然后才能开始信息的传输。在信息传输结束后，终端还必须发出拆连请求，网络释放连接。电话是一个典型的例子，只有在网络响应了振铃并接通线路之后，通话才能开始。通话结束，用户挂机后，网络才释放这条电路。在**无连接**的网络中，一个终端向另一个终端传送数据包并不需要事先得到网络的许可，而网络也只是将每个数据包作为独立的个体进行传递，例如分组交换中的数据报模式。以邮件的投递作为一个理解无连接的例子：人们并不需要向邮局作任何声明就可以投信；而邮局对每一封信都独立（与其他信无关联）地进行处理，并不关注是否还有其他信件投向同一个目的地址。

电路交换网络是面向连接的。连接可以通过呼叫动态地建立，也可以是永久性、或半永久性的专线连接。分组交换的网络则可分为面向连接的和无连接的两种：采用虚电路模式的，如帧中继、ATM 和 MPLS（见 7.8.3 节），属于面向连接的，而采用数据报模式的，如以太网、无线局域网（WLAN）和互联网则是无连接的。在面向连接的网络中，网络在建立连接时，能够（或有可能）为该连接预留一定的资源；当资源不够的时候，还可以拒绝接纳用户的呼叫，从而使 QoS 得到（或有一定程度的）保障。在无连接的网络中，由于网络“觉察”不到连接的存在，资源的预留就显得困难；不过，“无连接”也省去了呼叫建立所产生的延时。

7.3.3　资源预留、资源分配和资源独享

任何一个网络上总有许多对通信过程同时存在，它们以某种方式共享着网络资源。资源的管理与 QoS 保障有着密切的关系，现在我们从这个角度来区分不同的网络。

网络为某个特定的通信过程预留（Reserve）资源是指它从自己的总资源（如吞吐量、节点缓存器容量等）中规划出一部分给该通信过程，但是这部分资源并没有“物理地”给予该通信过程，网络只是通过资源预留来对自己的资源进行预算，以决定是否接纳新的呼叫。由于预留的资源并不等于通信过程所实际消耗的资源，“超预算”的事情很可能发生，因而通信过程的 QoS 也只是从统计

的意义上来说得到保障。这和我们用电话向航空公司预订机票类似:航空公司并未给顾客一个座位号,而只根据预约电话的多少、飞机可容纳的总人数、以及预定而不实际乘机的概率等因素给顾客一个大概的承诺。显然,顾客预订后得不到机票的可能性是存在的,但是,航空公司毕竟以某种机制在做座位预订的工作,能预订总比不预订要好。从统计的意义上来说,进行预约得到座位的可能性要比不预约大得多。

资源分配(Allocated)则比资源预留进了一步,它是把一部分资源实际分配给了某个特定的通信过程。但是,当网络发现该通信过程没有充分利用分配给它的资源,或者在网络发生严重拥塞时,可能动态地将部分已分配给它的资源重新分配给其他的通信过程。因此该通信过程的 QoS 保障可能是确定的,也可能是统计意义上的。这类似于我们在向航空公司预订机票时得到了一个确定的座位号,只要航空公司没有不小心把这个座位号给出去两次,登机时你可以放心一定会有你的座位。反之,如果你没有赶上飞机,航空公司也可能在飞机起飞前将你的座位分配给其他旅客。

网络在建立通信过程时就把一部分资源"物理地"划归该通信过程所有,并在该通信过程结束之前,不会将划归给它的资源让其他通信过程分享,也不会再重新分配给他人,这就是资源独享(Dedicated)的情况。此时,该通信过程的 QoS 是得到确定性保障的。在电路交换的网络中,分配给一对终端使用的带宽就是独享的。

如果网络既不给通信过程预留、也不给它们分配资源,只是利用自己的全部资源尽力而为地为所有的通信过程服务,那么,这些通信过程的 QoS 就与网络的负荷有关,也就是说,QoS 是没有保障的。这样的网络通常称为**"尽力而为"**(Best-Effort)网络,传统的共享介质的以太网和互联网都属于这种类型。

7.4 电路交换广域网对多媒体信息传输的支持

7.4.1 电路交换广域网

广域网(Wide-Area Network,WAN)是指跨越长距离、并需要使用干线传输系统和节点设备的网络。电路交换的广域网通常由电信部门运营。如前所述,电路交换网络的特点是:

(1) 是面向连接的,且在整个会话过程中独享固定的比特率;

(2) 传输延时由传播延时和所经过的中间设备的延时之和决定。ITU-T 规定每个同步 TDM 中间设备、或交换机的延时应小于 450 μs,因而传输延时一般在几毫秒到几十毫秒的量级,是比较小的;

(3) 仅存在物理延时抖动,可以认为传输延时是恒定的;

(4) 不支持多播。当需要多播时,必须加入 MCU。从前三个特点来看,电路交换网络适合于多媒体信息特别是连续媒体的实时传输,而且其 QoS 是得到确定性保障的。

在整个会话过程中,比特率得到保障,延时小,延时抖动小。这三个特征结合在一起称为**等时性**(Isochronism)。等时性是网络支持连续媒体流实时传输的保证。显然,电路交换网是具备等时性的。下面我们对几种电路交换广域网作一些具体分析。

公用电话网的骨干网建立在准同步/同步数字系列 PDH/SDH(Plesio-Chronous Digital Hierachy/Synchronous Digital Hierachy)的基础上,其本地回路使用双绞线传输,信道带宽较窄(3.1 kHz),而且是模拟的,多媒体信息需要通过调制/解调器(Modem)接入。调制/解调器的速率一般为 164 kb/s,可以支持低速率的多媒体业务,例如低质量的可视电话和多媒体会议等。近年来得到迅速发展的 xDSL 技术(见 7.10.1 节)使用户可以通过普通电话线得到几 Mb/s 以上的传输速率,但此时它是作为因特网的一种宽带接入方式,并不在电路交换的模式下工作。

N-ISDN 既可以经过交换机,也可以用专线方式提供业务。它的用户速率有如下几种:(1)基

本速率接口 BRI(Basic Rate Interface)：两个 64 kb/s 的 B 信道和一个 16 kb/s 的 D 信道，总共 144 kb/s；(2)一次群速率接口 PRI(Primary Rate Interface)：30 个 64 kb/s 的 B 信道和两个 64 kb/s 的 D 信道，总共 2.048 Mb/s(E1 接口)；(3)ITU-T 还允许在一次链接中，将连续的 DS-0(64 kb/s)信道合并在一起提供给用户。其模式有 H0：6 个 DS-0 信道，总共 384 kb/s；H11：24 个 DS-0 信道，总共 1 536 kb/s；H12：30 个 DS-0 信道，总共 1 920 kb/s；(4)近年来公布的 ISDN 多速率(ISDN Multirate)允许从 2～24 个 DS-0 信道合并的模式接入。N-ISDN的这些接口可用于中等质量或较高质量的会议电视。

DDN 提供永久、或半永久连接的数字信道，传输速率较高，可为 $n\times 64$ kb/s($n=1\sim 32$)。DDN 传输通道对用户数据完全“透明”，即对用户数据不经过任何协议的处理、直接传送，因此适于多媒体信息的实时传输。但是，在 DDN 网上无论开放点对点、还是多点的通信，都需要由网管中心来建立和释放连接，这就限制了它的服务对象只能是大型用户的租用。会议室型的电视会议系统常常使用 DDN 信道。

早期的蜂窝移动网(见 7.9.3 节)是电路交换的网络，但是发展到可以支持多媒体应用的第 2.5 代之后，已开始转向分组交换的网络。

7.4.2 多点控制单元

在只支持点到点通信的电路交换网络中，要实现 n 个用户之间的会议型服务，必须在每两个参与者之间建立一条双向的链路，共需 C_n^2 条链路。如图 7-4(a)中的 4 用户系统，需要建立 6 对线路才可能将每个参与者的声音和图像传送给其他的参与者。当 n 增大时，网络资源的浪费将很大。

如果在电路交换的网络中加入多点控制单元(MCU)支持多播的功能，则如图 7-4(b)所示，一个 4 用户系统，只需建立 4 对线路。此时，各用户终端的多媒体数据传送到 MCU，经过 MCU 的处理再返回各个终端。

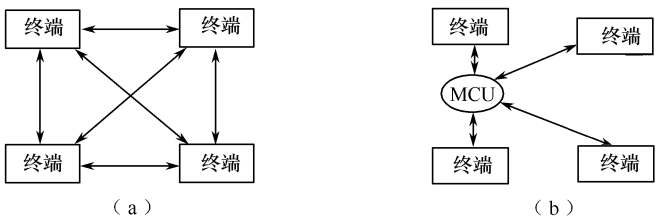


图 7-4 电路交换网络中实现多点通信的两种方式

MCU 也常称为多点控制服务器(MCS)，它的处理功能主要包括：

(1) 音频信号桥接。将各终端送来的音频信号混合，这通常称为桥接，然后再送回各个终端；或者从中选择出一路信号送给其他终端，这通常称为切换。选择的方式可以是轮流传送每一路信号(轮询)、固定只传送主会场信号，或由主席控制信号的切换等。

(2) 视频信号切换或混合。采取轮询、主席控制等方式将某一路视频信号送给各个终端；也可以采用声音激励的方式进行切换，将发言者的图像送给各参与者。近年来的 MCU 可以将多路图像组合成一个多窗口画面通过一个信道送给参与者。

(3) 数据切换。将会议中涉及的数据按与音频或视频信号类似的方法处理。

(4) 会议控制。对发言权、共享设备(如摄像机、白板)的控制，以及有关的通信控制(见 8.3.6 节)等。

一个 MCU 设备的输入/输出端口的个数是有限的，当与会者的数目超过端口数时，需要用多个 MCU 构成网络，图 7-5 给出一个两层结构的例子。MCU 的规模可按树形结构进行扩展。

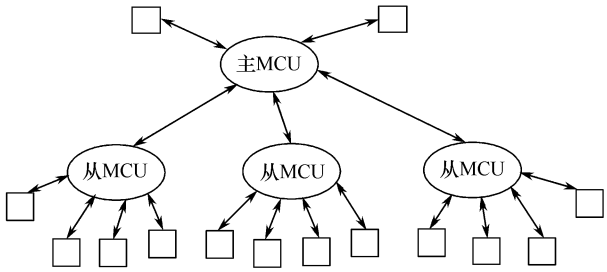


图 7-5　两级 MCU 构成的网络

7.5　分组交换广域网对多媒体信息传输的支持

本节将讨论若干电信部门运营的分组交换网络，互联(IP)网虽然也属于分组交换的广域网，但由于它在构建多媒体传输网络中的重要性，我们将在专门的章节中进行讨论。

7.5.1　帧中继

帧中继是面向连接的第二代数据传输网络。20 世纪 70 年代产生的第 1 代数据传输网 X.25 由于当时传输线路质量的限制，因而具有复杂的检错、纠错和流量控制机制，这使得它的传输速率低，延时长且抖动大。而现今的传输线路质量比 20 世纪 70 年代已有很大的提高，因此，整个差错检错和纠错处理可以从帧中继网中去除，差错控制主要在端点进行。此外，帧中继网也不对每个虚电路进行流量控制，仅在网络拥塞时可能给出一个粗略的指示。协议的简化使帧中继的传输延时比 X.25 网的要降低很多。

帧中继不直接与终端设备相连，它是针对局域网之间互联和多个信源的数据流(计算机数据、文件、电子邮件等)混合而设计的，带宽一般为 64 kb/s～2 Mb/s，在有些情况下，可能达到 140 Mb/s 左右。通常与它直接相连的设备是路由器、网关等。

初期的帧中继只允许建立永久性的虚连接 PVC(Permanent Virtual Connection)，而且对带宽和延时抖动没有什么保障，难以支持实时多媒体信息的传输。近年来，一些厂家在帧中继中引入了资源分配机制，以虚电路来仿真电路交换的电路，从而使带宽或延时抖动的限制得到了一定程度的保障。在帧中继中也有可能加入优先级机制，以便给予声音和图像数据流以高优先级，有利于它们的实时传输。还有一些厂家建议将实时数据的压缩和解压缩部件集成到帧中继设备中构成所谓的帧中继交换机。其具体做法是将压缩后的实时信号分割成数据帧，再在帧中继交换机中与 LAN、或其他方面来的数据复用，这就为实时媒体直接进入帧中继网提供了途径。以上可以看出，帧中继对多媒体信息传输的支持程度主要取决于实现它的具体环境和设施。

7.5.2　异步传输模式 ATM

异步传输模式 **ATM**(Asynchnous Transfer Mode)是 ITU-T 为**宽带综合业务数字网**(B-ISDN)所选择的传输模式，它的底层传输系统可以是 PDH 或 SDH。B-ISDN 是国际电联在 20 世纪 80 年代提出的概念，其目标是以一个综合的、通用的网络来承载全部现有的和未来可能出现的业务。但是，由于 B-ISDN 在许多方面，例如交换设备、终端设备、传输模式和用户接入等，与旧有的通信系统有较大的不同，而整个系统的改造并非易事，因此，尽管国际电联为其制定了一系列标准，B-ISDN 并未得到预期的发展。不过 ATM 作为一种高速包交换和复用技术在构建广域网和 xDSL 宽带接入配置中仍然广泛使用，而且它的许多概念被近年来出现并获得重要发展的多协议标记交换 MPLS(见 7.9.3 节)的标准所采纳。

1. ATM 原理

ATM 采用固定长度的数据包，称为**信元**。虽然选用长度固定的信元，在有些情况下（如传送几个字节的短消息时）会因信元填充不满而有所浪费，但信元长度固定有利于快速交换的实现，以及纠错编码的实施。另一方面，长度大的包由于附加信息（包头）占的比例小而效率较高，但是在节点逐级存储—转发的过程中，整个包必须完全被接收下来之后才能转发，从而导致延迟增长。此外，长度大的包如果丢失，信息损失肯定比长度小的包要多。考虑到上述种种因素的折中，ATM 信元长度确定为 53 字节，其中 5 个字节为信元头，48 个字节为数据。信元头中包括：**虚通道**和**虚路径**的标识符，虚通道（Virtual Channel）和虚路径（Virtual Path）是 ATM 的两种虚连接方式；数据类型域用来标识信元所携带数据的类型；信元丢失优先级域标识在网络拥塞时，该信元被丢弃的优先程度；通用流量控制域是为流量控制的需要而准备的；错误检测域则用于信元头误码的检测和校正等。

ATM 是面向连接的网络，终端（或网关）通过 ATM 的虚通道相互连接。两个终端（或网关）之间的多个虚通道可以聚合在一起，像一个虚拟的管道，称为虚路径。这就是说 ATM 允许在一个连接中建立多个逻辑通道，这便于具有不同 QoS 要求的媒体采用不同的通道传输。

ATM 具有高吞吐量、低延时和高速交换的能力，信元在硬件中交换。当发送端和接收端之间建立起虚通道之后，沿途的 ATM 交换机直接按虚通道传输信元，而不必像数据报分组网的路由器那样，利用软件根据每个数据包的目的地址寻找路由。ATM 继承了分组交换网络中利用统计复用提高资源利用率的优点，几个信源可以被结合到一条链路上，当其中一个信源发送数据的速率低于它的平均速率时，它所剩余的带宽可为该链路上的其他信源享用。同时，ATM 采用的统计复用允许某一数据流瞬时地超过其平均速率，这对于突发度较高的多媒体数据是很有利的。ATM 与一般的分组交换网络有所不同的是，它有一定的措施防止由于过多的信源复用同一链路、或信源送入过多的数据而导致网络的过负荷。换句话说，ATM 网具有对流量进行控制的功能。ATM 流量控制功能中最基本的两项为**连接接纳控制**（CAC）和**使用参数控制** UPC（Usage Parameter Control）。CAC 根据网络资源决定接受、或者拒绝用户的呼叫；UPC 对信源输出速率是否超过约定值进行监测和管理。ATM 的流量控制对用户的 QoS 要求得到统计性的保障有着重要的意义。

2. ATM 服务类型和 ATM 适配层

ATM 有定义明确的服务等级，也就是说，它支持对定性描述 QoS 的保障。ATM 的服务类别由图 7-6 所示。第一类称为 CBR 服务，它提供带宽固定、延时确定的服务。由于它与电路交换信道的性能相近，因此，常称为电路仿真模式。此类服务适合于电话，以及恒定速率的实时媒体的传输。当信源要求 CBR 连接时，它必须将它的峰值速率通知网络，这个速率在整个通信过程中都为该信源使用。第二类称为实时 VBR（VBR-RT）服务，它提供延时确定、带宽不固定的服务，特别适合于经压缩编码后的声音或视频信号的传输。当信源要求 VBR 连接时，它需要通知网络它的平均速率、峰值速率和突发的最大长度（峰值速率的持续时间）等参数。第三类称为非实时 VBR 服务，它适合于没有延时要求、而突发性强的数据传输。与第二类服务一样，它需要通知网络它的平均速率、峰值速率和突发最大长度等参数。前三类服务均能提供限定信元丢失率的保障。第四类为 UBR（Unspecified Bit Rate）服务。它不提供带宽、延时和信元丢失率的保障，适合于对信元丢失有一定容忍程度的应用。在连接建立时可以提出、也可以不提出对峰值速率的要求。第五类为 ABR（Available Bit Rate）服务，它与第四类相似，只是网络能在拥塞时向信源反馈信息，从而使信源能够适当降低自己的输出速率。ABR 即使在拥塞时也能保障最小的带宽，但没有延时的保障。ABR 的主要目的在于将网络闲置的带宽利用起来，它仿真 LAN 的无连接方式，这使得利用 ABR 通过 ATM 的 LAN 互联变得和通过路由器互联的方式一致。



图 7-6　ATM 服务等级

为了支持不同的服务，如图 7-7 所示，ATM 在底层（物理层和 ATM 层）与高层协议之间加了一个 ATM 适配层 AAL（ATM Adaptation Layer）。图中，物理层用来传输比特（信元）流；ATM 层完成交换、路由选择和复用；AAL 层则只在终端上存在，它负责将不同服务的信息流按照 4 种 AAL 协议中的一种打包封装成信元流。AAL 的每一种协议定义一种 AAL 头，AAL 头占用信元用户数据（48 B）的一部分位置。AAL 1 用于支持 CBR 业务，它占用一个字节传送 AAL1 的头，其余 47B 传送上层传递下来的用户数据。AAL 2 用于支持 VBR-RT 业务。AAL 3 和 AAL 4 在发展过程中逐渐趋于合并成一个，称为 AAL 3/4，用于支持面向连接的或无连接的突发数据业务。由于 AAL 3/4 复用和它的复杂协议在许多应用中并不需要，从而产生了为面向连接的数据传输而设计的 ALL 5，它的开销较小、检错较好。AAL 5 不需要再附加 AAL 头，只将上层传递下来的用户数据单元加上 8B 的“尾”和一定的填充字节凑成 48B 的整数倍，然后分割成信元传送。由于 AAL 5 的简单有效，它被应用于 TCP/IP 数据和低造价的实时媒体的传输。



图 7-7　ATM 协议模

7.6　局域网对多媒体信息传输的支持

局域网（Local Area Network，LAN）是计算机网络中的概念，它是指在一个建筑物内或一个园区内的独立的计算机网。传统的局域网包括**以太网**（Ethernet）、**令牌环**（Token Ring）和 **FDDI**（Fiber Distributed Data Interface）。由于的年来网络技术的迅速发展使得以太网成为最重要的局域网，大量的多媒体终端设备都支持以太网接口；同时，以太网也成为颇具潜力的城域网甚至广域网技术，因此，本节重点介绍以太网的发展及它对多媒体信息传输支持。

7.6.1　共享介质的局域网

1. 局域网通信的特点

传统局域网的共同特点是将所有的终端（在计算机领域中常称为**站**）都连接到一个共同的传输介质（例如一条同轴电缆、一对双绞线、或一根光缆）上。局域网的几种常见拓扑结构是总线、树形、环形和星形（见图 7-8）。其中总线是树形的特殊形式，它只有一个不带树枝的树干。在现代局域网中，星形拓扑占主要地位，其中连接各个站的中心设备称为集线器。

出于安全性和可靠性的考虑，可能不希望将太多站连到一个传输介质上；而且局域网的性能会随着站数目的增加或传输介质长度的增加而降低。此时，可以将共享的介质分成段，每段连接一定数目的站，称为网段。网段之间用桥、或路由器相连。

公共介质构成了连接在它上面的多个站之间的相互通信的通道，但是由于传统 LAN 采用基带信号进行传输，当一个通信连接占有传输介质时，另一对站之间就不能进行通信。为了实现多对站之间的同时通信，发送信息的站将数据流分成段（称为帧），在各自占有传输介质的时间间隙内，逐段将数据发送出去。这实际上是一种异步时分复用的模式。发送站在何时取得介质的占有权取决于一套预先制定的规则，称为**介质访问控制** MAC（Medium Access Control）协议，协议的不同就

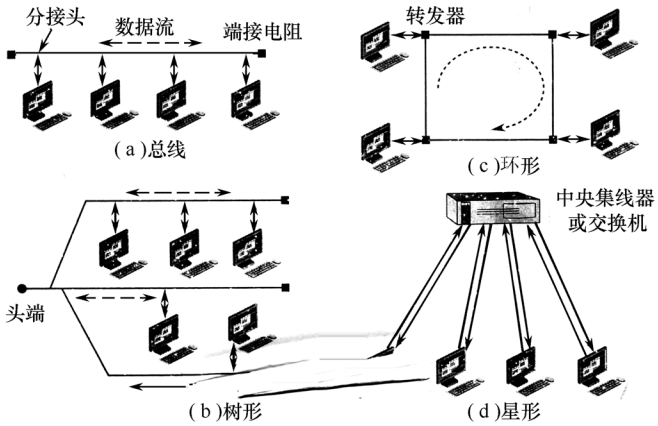


图 7-8　局域网的拓扑结构

形成了不同类型的 LAN。

2. 以太网

以太网的 MAC 协议称为 **CSMA-CD**(Carrier Sense Multiple Access with Collision Detection)协议,由 IEEE 802.3 和 ISO 8802.3 标准所规定。任何一个站在欲进行数据发送之前,先通过检测总线上的信号获知总线是否空闲,如果空闲,则可发送一个数据帧(数据帧产生的电信号沿总线向两个方向传输);如果不空闲,则该站可以等待一个随机的时间,再行检测,或者一直连续不断地检测,直至总线空闲便立即发送。由于检测是各站各自分别进行的,有可能多个站同时检测到总线空闲而同时发送数据,这些数据发生"碰撞"而在总线上形成非正常的电平。首先检测到"碰撞"的站将通过总线向其他站告警。总线上所有的站都能收到告警信号,正在发送数据的站则立刻停止发送,各自等待一个随机的时间之后再重新尝试数据的发送。同时,为了避免网络负载的进一步加重,等待时间的长短随尝试次数的增加而按一个称为**截断二进指数回退**(Trancated binary exponential backoff)的算法呈指数增长[2]。相互"碰撞"的数据作废需要重发,在很多站都要进行数据发送时,"碰撞"可能连续发生,因此存在着数据完全发送不出去的可能性。

连接以太网网段的桥和路由器以存储—转发的方式工作。当数据需要跨越网段时,路由器(或桥)必须首先将整个数据帧接收下来,并进行某些处理后,再发送到另一个网段上,这显然要引入一定的延时。

传统以太网的总线吞吐量为 10 Mb/s,后来发展到 100 Mb/s,称为快速以太网。

3. 传统以太网的性能

假如一个 10 Mb/s 的以太网段上只有一对站在相互进行通信,总吞吐量可达 9 Mb/s;如果有 100 个站同时在发送平均长度为 1 kb 的数据帧,该网段的实际总吞吐量只能达到其最大值的 36%,即 3.6 Mb/s。[1] 100 Base-T 以太网将吞吐量提高了 10 倍,但由于其 MAC 协议仍然是 CSMA-CD,每个站所享有的吞吐量还是受网络负荷的影响。

减少一个网段上的站数,甚至于一个网段上只连接一个站,是提高单个站点享有带宽的一种解决方案。但是要使一定数量的站点同在一个网上工作,就需要用若干个路由器(或桥)将各网段连接在一起。此时,路由器或桥将成为系统的瓶颈。同时,路由器上的存储—转发工作模式将增加传输延时,这对实时多媒体数据的传输也是不利的。

以太网的传输延时很大程度上取决于得到介质访问权所需要的等待时间。网络负荷增大时,"碰撞"的可能性增大,等待时间加长。多媒体数据传输所要求的对延时时间和延时抖动大小的限制难以在这样的网络上得到保障。

所有共享介质的 LAN 支持帧一级的广播和多播。LAN 实质上是一个广播型的网络，每个站都具有从公共介质上接收所有信息的能力。如果数据帧前面所加的地址是特定的广播地址，则每一个站都知道自己应当接收这个信息。多播也可用类似的方式实现，这时加在数据帧前面的地址是“组地址”，即属于该组成员的站接收这一信息。但问题是，端站的网络接口往往不能区分本站是否属于该组，它只能把信息接收下来交由 CPU 去分析、取舍，这个过程要消耗 CPU 资源，实际上与广播模式没有什么不同。因此，LAN 上的多播功能是有局限性的。

4. 以太网帧交换

共享介质的特点造成了前面讨论的许多缺陷，人们开始考虑将交换的概念引入局域网。用交换机代替图 7-8(d)中的集线器，数据就可以通过交换机直接到达信宿的端口，而不必像使用集线器那样需要经过每一个端站。

以太网交换机的使用并不改变传统端站的网络接口，但它部分地改变了传统 LAN 共享介质的特点，它通过交叉连接可以支持多对(取决于交换机的容量)端站间的同时通信。因此，可以在一个网段上只连少数端站、甚至于一个端站(通常是服务器)，以保证多媒体数据传输所需要的吞吐量；同时，用以太网交换机代替桥或路由器连接各个网段，缓解了由路由器产生的瓶颈和过长的时延。

以太网交换机能够处理长度不固定的以太网帧，并支持帧一级的多播。由于交换在数据链路(MAC)层进行，因此以太网交换机也称为**第二层交换机**。

7.6.2 吉比特以太网

吉比特以太网继承了传统以太网的许多特性，如帧结构、最小和最大帧长，以及高层协议等，使得原来在 10 Mb/100 Mb 以太网上开发的应用可以无需改变地在吉比特以太网上运行。但是在另一方面，吉比特以太网并不是传输速率上的简单升级，它在数据链路层和物理层上的技术变革，改变了传统以太网共享传输介质所引起的传输效率低和传输距离短的局限性，使得吉比特以太网可以作为 LAN 的骨干网、城域网、甚至广域网技术而应用在更大的地域范围上。本节将着重介绍后一方面的情况。

1. 系统结构

一个吉比特以太网站点的系统结构如图 7-9 所示。从图上可以看出，吉比特以太网是只涉及数据链路层和物理层的技术，只要设备驱动符合标准的网络驱动接口，高层的各种协议，如 TCP/IP 等，并不因底层网络技术的变化而受影响。吉比特以太网的网络控制器包括介质访问控制实体、线路编码/解码和驱动器/接收器 3 个部分，其中线路编码/解码的作用是将数据转换成适合于传输介质传输的形式；驱动器/接收器是传输介质所要求的发送和接收装置，例如对于铜导线来说可能是电子装置，对于光缆来说可能是激光器和光检测器。

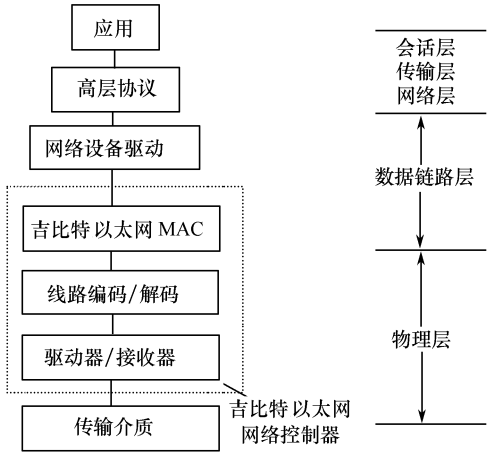


图 7-9　吉比特以太网系统结构

2. 全双工工作

吉比特以太网以**全双工**方式工作。为了对它的特点有比较清楚的了解，让我们回顾一下传统以太网的工作方式。

我们知道传统以太网的传输介质(铜缆)是由多台主机共享的。对于某一对主机而言，无论数

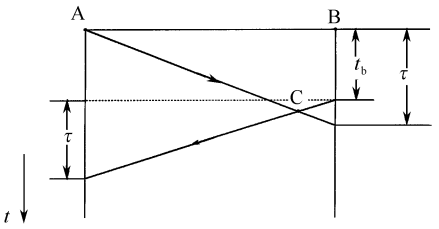


图 7-10　传统以太网信道冲突时间图

据的发送还是接收，都经过同一条传输介质，因此收、发不能同时进行，这称为**半双工**工作。对于多台主机而言，当一台主机占用信道时，其他主机不能使用该信道，多台主机争用信道的冲突问题由以太网的 MAC 协议 CSMA/CD 来解决。

使用 CSMA/CD 协议使以太网的传输距离和数据帧的长度受到一定的限制，现在以图 7-10 所示的例子加以说明。假设图中 A、B 两个主机之间的单程传输延时为 τ。在 A 开始发送一个数据帧之后的 t_b（$t_b<\tau$）时刻，B 欲进行发送，由于 A 所发送的数据尚未到达 B，B 通过监听认为信道是空闲的，于是进行发送。A、B 的数据帧在 C 点“碰撞”，而 A 和 B 则分别要到（$\tau+t_b$）和 τ 时刻才知道已经发生“碰撞”，并各自发出“告警”信号。当 t_b 接近于 τ 时，A 监听到“碰撞”的时间约为双程传输延时 2τ。根据 CSMA/CD 协议，主机在边发送时边监听，如果监听到信道发生冲突，双方停止发送，“碰撞”的帧需要稍后再重发。假设 A 在检测到冲突时数据帧已发送完毕，A 将无法判断是否是自己的帧受到“碰撞”，也不会再重发。因此要保证冲突检测的正常进行，最短的以太网帧的传输时间必须大于最坏情况下（A、B 在网的两个最远端）的双程传输延时 2τ。这同时也说明了对传输距离的限制。

以太网交换机在很大程度上改变了传统以太网共享传输介质的状况。如果交换机的每个端口上只连接一台主机，主机通过两对双绞线与交换机相连，可以构成一个支持同时进行发送和接收的全双工信道。不过由于标准的以太网接口（网卡）是按半双工工作设计的，它将本机的边发送边接收认为是“冲突”，因此必须对接口作一些改造才能实现全双工工作。所幸的是，接口无需加入任何功能，而只需要中止某些功能，如载波监听、冲突检测和收、发信道间的回路等，就可以完成这一改造。由此可以看出，全双工工作的以太网不需要 MAC 协议，它与半双工网相同的部分只是数据帧结构以及传输介质所需的线路编码方法。

由于吉比特以太网通常工作在全双工方式下，此时传输距离不受 MAC 协议限制，因此它能够发展成为一种城域网或广域网技术。

3. 吉比特以太网标准

吉比特以太网的出现使得以太网这种简单、便宜、能与 TCP/IP 一起很好工作的技术可以从终端一直延伸到广域网。协议一致的统一网络将提高光缆带宽的利用率，减少管理的开销。与 ATM 和 SDH 相比，吉比特以太网不需要那样严格的时钟同步要求，但是它在可靠性和 QoS 支持方面尚有所不及。

IEEE 已为 1Gb、10Gb 和 40G/100G 以太网分别制定了 802.3Z、802.3ae/802.3aq/802.3ak/802.3an 和 802.3ba 标准。短距离连接使用双绞线和多模光缆，以点到点的方式提供站内服务器机群和交换机之间的互联；长距离连接使用多模或单模光缆，在城域网或广域网中应用。

7.6.3　以太网 QoS 保障

1. 流量控制

传统的以太网没有流量控制，进行通信的两台主机通过高层协议（如 TCP）来保证数据的发送速度不高于接收端的接收能力和不造成接收端缓存区的溢出。在以太网中引入以太网交换机以后，交换机节点无法使用上述办法控制流量。当数据率较高、交换机端口较多时，交换机缓存器可能在某些时刻溢出，造成数据帧的丢失。当数据率高达吉比特时流量控制就显得更为必要。

IEEE 802.3x 为全双工以太网规定了流量控制的方法，该方法只适用于点到点的链路，即两个

主机之间、一个主机与一个交换机、或两个交换机之间的链路。该方法可以对两个方向的链路进行流量控制，这称为对称式的；也可以只对一个方向的链路进行流量控制，这称为非对称式的。吉比特以太网的流量控制在自动协商过程中进行配置。

2. 包分类服务

近年来制定的 IEE 802.1p 为以太网规定了包(以太网帧)分类的方法，以便交换机根据包的优先级分别进行处理，使以太网具有 MAC 层的 CoS 能力。

IEEE 802.1p 利用 IEEE 802.1q 插入的标签来表示优先级。IEEE 802.1q 是关于虚拟局域网(VLAN)的标准。支持 IEEE 802.1q 的主机在发送数据帧时，在原来以太网帧头中插入一个 4B 的标签头，图 7-11 表示出标签头所处的位置及包含的域。其中 TPID 是 IEEE 802.1q 标签的标识，CFI 用于识别是否是标准的以太网 MAC 地址，VID 是 VLAN 的标识符。IEEE 802.1p 利用标签头中 3b 的优先级域来标识 8 个服务类别。

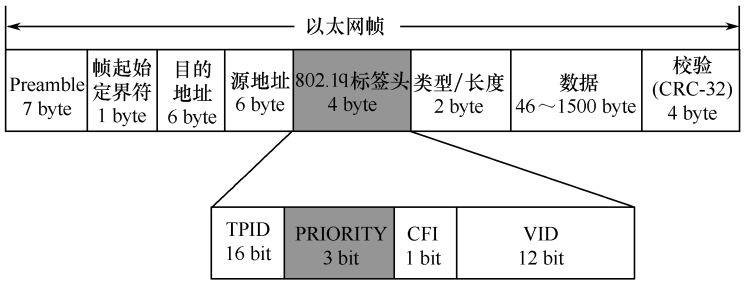


图 7-11　802.1q 标签头

对这 8 个优先级，IEEE 给出了一个宽泛的定义(非强制性)：最高级 7 用于网络关键业务，如路由协议的路由表更新；优先级 6 和优先级 5 用于对延时敏感的业务，如交互视频与音频；优先级 4 到优先级 1 可分别分配给从“控制负荷”(Controlled load)业务(如流媒体或企业关键信息)到“允许包丢失”业务的不同类别业务；优先级 0 作为默认，用于尽力而为的业务。

7.7　IP 网对多媒体信息传输的支持

IP 网指使用一组称为因特网(Internet)协议的网络。因特网是由 1969 年开始的美国国防部的研究网络 ARPAnet 发展而来；20 世纪 80 年代出现的著名的 TCP/IP 促进了该网络的发展，使其连接范围扩展到大学、研究单位、政府机关、公司等机构；20 世纪 90 年代中期出现的 World Wide Web 技术进一步推动了因特网的迅速发展，使之演变成为一个世界范围内的、最具影响力的信息网络。近年来针对在 IP 网上提供多媒体服务所存在的问题，各国科学研究和工程技术人员进行了大量的工作，使其性能已经、并继续得到改善，再加上它固有的简单性和开放性(独立于它的上层和下层协议)，IP 已经成为电信网与计算机网融合中网络层的事实上的标准。

7.7.1　传统的 IP 网(IPv4)

1. 网际互联的原理

单个网络内的资源往往无法满足用户的需求。为了让用户能够利用所在网络范围之外的资源，就需要将各种不同类型(如使用不同传输介质和/或不同通信协议)的网络互相连接起来，使得位于任何成员网络(或称为子网络)上的两个站点之间能够互相通信。TCP/IP 较好地解决了这一问题。

IP 协议由推动因特网技术发展的组织 **IETF**(Internet Engineering Task Force)提出的建议 RFC 791 所规定。IP 协议采用无连接的分组交换方式(数据报)来实现网际互联,子网之间的连接设备称为路由器。图 7-12 给出了两个子网互联的例子,其中(b)为(a)中设备对应的协议栈。上层的用户数据在 IP 层被分割打成 IP 包(或称数据报),并加上一个固定 20B 和一个可变选项字段的包头(见图 7-13)。IP 允许的包长最大至 64kB,但太大的包长可能引起延时过长和拥塞问题,一般采用的包长是 1500B,因为这能适应以太网 1536B 的典型帧,且有利于降低重传时间和提升高速链路的性能。

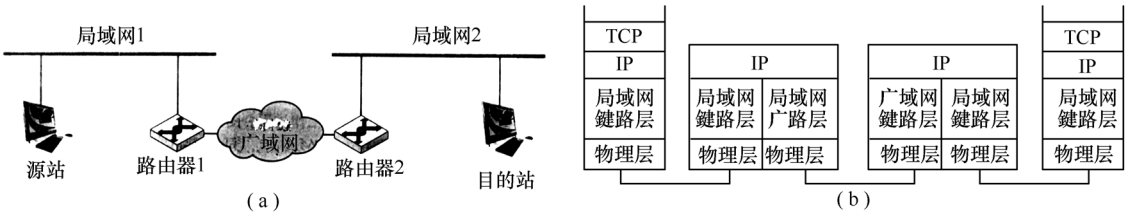


图 7-12　子网的互联

路由器根据每个 IP 包头中的目的地址为其选择路由,IP 包从 IP 网的一个路由器跳转到下一个路由器,直至到达目的子网。由于在每个路由器上都必须要进行下一跳路由的选择(每个包都是独立的),因此同一源和目的端站之间,不同的数据包可能途径不同的路由。路由器采用存储一转发机制传递数据包,并进行底层协议的转换。例如在图 7-12 中,源站的 IP 接收来自上层(如 TCP)的需要发送的数据块,将其加上 IP 头部构成 IP 数据报,然后这个数据报通过局域网协议封装成帧,并发送至路由器 1。路由器 1 剥掉接收到的局域网帧的封装以读取 IP 数据报,然后将该数据报按广域网协议波段进行封装,并通过广域网送至路由器 2。路由 2 器剥去广域网的字段,将其恢复成数据报,然后再用适合于局域网 2 的协议字段封装,并将其送到目的站。我们看到,这种方式可以在不同类型的网络上向用户提供一致的通信服务。

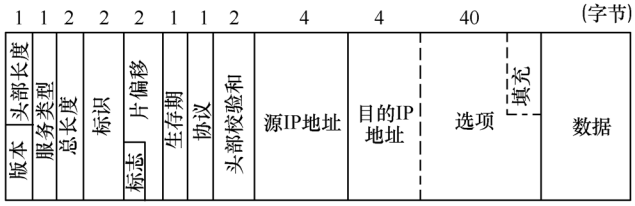


图 7-13　IPv4 数据报格式

2. 传统 IP 网的性能

由于 IP 网是由研究机构而非电信运营部门设计的网络,所以它着重于效率而不是可靠性。它要求协议简洁、速度快、尽可能地与底层具体传输系统的性能无关,对于传输中的差错,如丢包、包次序颠倒等,则留给终端去解决。传统 IP 网设计中也未考虑 QoS 保障和计费等问题。

IPv4 是一个无连接的、"尽力而为"的分组交换网络。由于路由器采用存储转发机制工作,又没有资源预留机制,因而传输延时会存在抖动。在网络负荷过重时,传输延时的变化可能达到秒的数量级;同时路由器可能由于其存储缓存器溢出而丢弃数据包并且不通知终端(有的路由器可能向信源传送一个网络暂时拥塞的指示,称为 Source-Quench)。此外,由于路由器为每一个 IP 包独立地选择路由,因此到达接收端的包的顺序不一定能得到保证。

上述问题的存在使得传统 IP 网上多媒体传输的带宽和延时抖动等要求都得不到保障。

7.7.2 IP 多播

1. 多播地址与多播组

IP 多播是 IP 的扩展功能。IP 网通过接收地址的格式来区分一般 IP 数据包和多播 IP 数据包。如果 32 位 IP 地址的前 4 位是“1110”，说明它是一个多播数据包，这类地址称为 D 类地址，地址的后 28 位是多播组的标识。

在 IP 多播中，组成员的状况可以是动态的，也不需要有主席(Central Authority)，任何终端可以在任何时间加入或退出任何一个组。这也称之为**开放式**的组。组内的终端可以是只发送的、只接收的、或者又发又收的。同一个终端还可以同时处于几个不同的组之中，图 7-14 给出了一个这样的例子，图中 A/V 终端向 B 组发送视频信号，同时向 A 组发送音频信号。

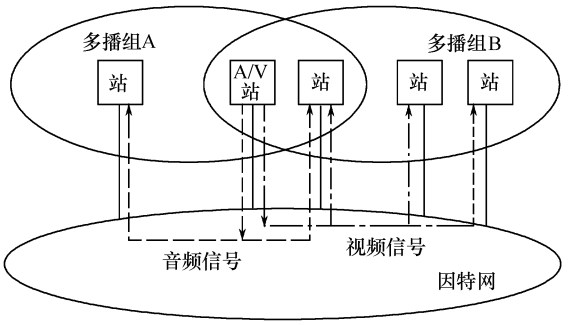


图 7-14　同一终端在不同多播组中的例子

如果一个主机想发起一个会议，首先需要由它的应用给出一个多播地址，或者通过多播地址动态用户分配协议 **MADCAP**(Multicast Address Dynamic Client Allocation Protocol)向 MADCAP 服务器申请一个多播地址；然后通过一定方式，例如通过**会话通知协议 SAP**(Session Announcement Protocol)，或在称之为会话目录(Session Directory)的目录下，公布这个地址以及会议的起始时间和大概的持续时间等，欲参加者则可根据这些信息在会议进行中随时加入。

2. 因特网组管理协议(IGMP)

因特网组管理协议(Internet Group Management Protocol)，IGMP 用于终端与距它最近的本地多播路由器间的联络，它由 RFC 1112 所规定。

多播路由器，如图 7-15 中的路由器 1，周期性地向本地子网上所有主机发送 IGMP Query 消息，要求所有主机报告它们当前是哪个组的组成员；每个主机则必须为每个它为成员的组返回一条独立的 Report 消息。值得注意，多播路由器并不需要详细掌握每个组的组成员名单，因为只要在它所属的子网内有一个组成员返回了 Report 消息，它就需要向该子网传递这个组的数据报文。如果多播路由器没有收到任何 Report，则说明本子网上该组的所有成员都已退出会议，路由器不再转发该组的报文。

从以上过程可以看出，一个主机退出一个多播组并不需要通知多播路由器；而参加一个多播组则需要立即向本地多播路由器发送一个 Report 消息，而不是等待路由器的 Query 消息之后再发。这样可以保证当此主机为本地网上该组第一个成员时，能及时地收到该多播组的信息。不过在 IGMP 的新版本中，为了减小“退出延时”，退出的主机也要向所有(如果有一个以上的话)本地多播路由器发送一条 Leave 消息，并给出退出的组地址。收到这条消息的路由器则针对该地址发送一个特定组的 Query 消息。如果没有返回相应的 Report，说明刚才退出的主机是本子网中该组的最后一个成员，路由器可以停止转发该组的报文。

3. 多播路由协议

多播路由协议使路由器具备多点路由的功能，即识别节点的多播组并寻找到达组内所有成员的最小代价的路径，此路径的拓扑常称为多播树。多播树应具有较低的端到端的延时，能够长期生存和易于扩展，并能支持动态成员。多播路由协议可以分为**域内**多播协议和**域间**多播协议两大类。域内多播协议实现数据包在域内的多播，其难点在于组内成员动态地改变，以及尽可能减少网络负载并避免环路路由的产生。典型的算法有 DVMRP(Distance Vector Multicast Routing Protocol)、MOSPF(Multicast Open Shortest Path First Routing)、CBT(Core-Based Tree)和 PIM(Protocol Independent Multicast)等[3]。域间多播协议完成数据包在域间的多播功能，其难点是选择一条最佳的域外链路把数据包送至位于域外的主机。典型的算法有 MBGP(MultiProtocol Border Gateway Protocol)、MSDP(Multicast Source Discovery Protocol)和 BGMP(Border Gateway Multicast protocol)等[3]。

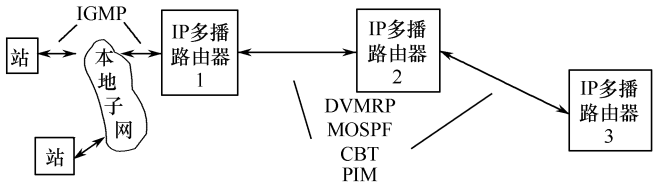


图 7-15　IP 多播

4. MBone

由于一般的因特网路由器并不都支持多播功能，而且传统因特网的运营者也由于计费的困难对提供此项业务不甚积极，因此支持多播的局部网络形成了“孤岛”。所谓的 **MBone**（Virtual Internet Backbone for Multicast IP，简称 Multicast Backbone）是一个将多个多播子网（“孤岛”）连接成一个大网的虚拟多播网络，两个多播子网之间通过 IP **管道**技术建立逻辑链路。图 7-16 给出了 MBone 的示意图。建立管道的具体的做法是，将链路一端的含有多播地址的 IP 报文装入另一个 IP 报文的用户数据字段中，这个外层 IP 报文的目的地址设为链路另一端目的多播路由器的众所周知的 IP 单播地址。通过这种方式，多播报文就可像单播报文一样，穿过不支持多播的网络到达逻辑链路的另一端。在目的多播路由器上，报文被恢复成多播报文进入另一个多播子网。

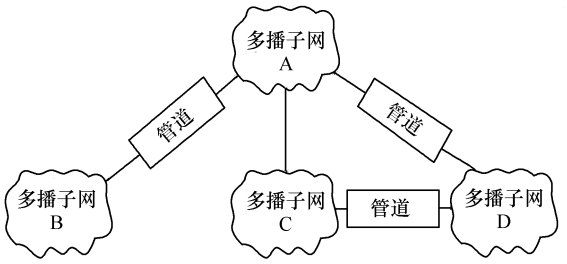


图 7-16　MBone

7.7.3　新一代 IP(IPv6)

IETF 于 1995 年年底公布了 RFC 1752，名为下一代的 IP，也称为 IPv6。图 7-17 给出了 IPv6 数据报的格式，其包头长度为 40B。

从多媒体信息传输的角度而言，IPv6 与 IPv4 相比有如下特点：

1. 地址空间和包头

如图 7-17 所示，在 IPv6 中，地址的位数升至 128 位，地址空间增加了 2^{96} 倍！这不仅为因特网的发展留下了足够大的空间，而且可以在构造地址时使用更多的网络结构层次，以减少路由表的大小和寻找路由的时间。虽然 IPv6 的包头比 IPv4 长(IPv4 包头为 20 字节)，但域的数目比 IPv4 包头少，因此路由器对 IPv6 包头进行处理所需的时间较少。

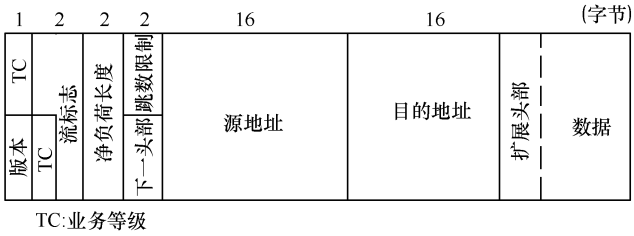


图 7-17　IPv6 数据报格式

2. 任播地址

除去单播和多播地址外，IPv6 增加了一种用于提高寻径效率的特殊单播地址，称为**任播地址**。一组接口（例如某个网络服务提供商所属的一组路由器）可以具有同一个任播地址。当一个节点欲向该地址传送数据包时，根据寻径协议度量“距离”的准则，数据包将传送给这组接口中“距离最近”的那一个。在某些应用场合任播地址是十分有用的。例如，给予分布在网络边缘的具有相同数据库的一组服务器（如 VOD 视频服务器）同一个任播地址，当用户请求点播时，该请求会自动送交给距离用户最近的服务器。

3. 包长

IPv6 只允许信源对消息进行分段。如果段长（包长）大于传输路径上允许的**最大传输单元 MTU** 的长度，路由器不像 IPv4 那样再对包进行拆分，而只简单丢弃，然后向信源传送一个出错消息。因此信源通常将消息分成最大为 576B 的段，这是多数网络支持的最小 MTU 的大小；或者通过“Path MTU discovery”程序事先了解传输路径上 MTU 的大小。路由器不对包进行拆分，则有利于提高它的传输速率。

4. QoS 能力

IPv6 包头中的**业务等级**（Traffic Class）域和**流标志**（Flow Label）域可以用来区分包的不同等级。

业务等级域（优先级域）与 IPv4 中的服务类型域（见图 7-13）类似，可以将从一个信源发出的数据包分为源提供拥塞控制的和源不提供拥塞控制的两种类型，每一种类型又可以分为 8 个优先级。所谓提供拥塞控制是指信源能对拥塞做出反应，例如 TCP（见 9.2.1 节）在收不到接收端发回的确认信号时，信源将重发信息，因此延时抖动较大。而源无拥塞控制的业务比较适合于实时数据的传输。在两类 16 个优先级中，级数高的优先级高，其中 0～7 用于允许尽力而为服务的应用，8～15 用于涉及实时媒体的应用。协议建议了不同应用对应的优先级，例如压缩视频比不压缩的音频的优先级高等。

流标志域定义具有相同传输服务要求的数据包序列。我们曾经用流（Stream）描述具有时间约束的连续媒体数据，而这里所说的流（Flow）是指从某一信源到某一（单播或多播）目的地的一个数据包序列，信源希望沿途的路由器对该序列中的数据包作相同的处理。为了区别起见，我们将前者称为**媒体流**，而后者称为**业务流**。业务流可以由连续媒体形成，也可以由连续传送的文件、数据或控制消息形成。

流标志域与业务等级域密切相关。对于尽力而为的业务，该域置为 0；对于无拥塞控制的业务，由发送端设定流标志，但须以某种方式（例如通过信令协议、或段间可选扩展包头）告知沿途的路由器对具有该标志的数据包应作何种特殊处理。这些处理可能涉及路径、资源分配、丢弃准则、计费和安全等方面，例如将数据包放进特殊的队列，以加快处理速度等。沿途的路由器则根据每个

包头中的流标志、源地址和目的地址识别特定的流，并结合指定的业务等级来保障该流需要的 QoS。

5. 安全体系

IPv4 除了在包头中有一个可选的安全标志域之外，没有什么安全机制，有关安全的措施只能在应用层实现。IPv6 能够提供 IP 层的安全机制，容易在此基础上建立多层安全网络和虚拟专用网。

7.8 IP QoS 保障机制

传统的 IP 网不能满足多媒体信息传输的要求，但近年来通信技术的迅速发展已经极大地改善了这种状况。在一方面，高速路由器和标记交换的引入，以及先进的宽带传输技术，例如**密集波分复用** DWDM(Dense Wave-Devision Multiplexing)，甚至**全光网络**的出现，使得 IP 核心网的带宽大幅度地提高。在另一方面，IETF 近年来又提出了几种 IP 网服务模型和 QoS 机制，为不同类型媒体的有效传输创造了条件。本节将讨论这后一方面的内容。

7.8.1 综合服务模型与资源预留协议 RSVP

IETF RFC 1633 提出了一个**综合服务模型**(Intergrated Service Model，**Intserv**)，其基本思想是通过资源预留来实现 QoS 保障，因此该模型的核心部分为一个**资源预留协议 RSVP**(Resource Reservation Protocol)。与 RSVP 和综合服务有关的过程由 RFC 2205－2216 规定。RSVP 是一种无连接方式的方法，保持着与 IP(以及 IP 多播)的兼容性。

在综合服务模型中除了“尽力而为”的服务之外，又引入了两类服务等级。一类属于虽然不能有确定性保障、但能提供与轻负荷时尽力而为服务相同等级质量的服务，可以用于具有数据调整能力、允许较小丢包率和排队延时的多媒体应用。这类服务称为**负荷受控(Controlled-Load)服务**。另一类用于有实时要求的应用，如声音和视频信号的实时传输。在这类应用中希望延时和延时抖动有预定的限制范围，以及带宽有一定程度的保障。这类服务称为**可保障(Guaranteed)服务**。

RSVP 为每一条单播或多播业务流预留资源，一条流由源和目的 IP 地址及(可选)传输层端口号所定义。RSVP 的一个显著特点是在无连接的网络上，实现了由接收端启动的资源预留。在多点会议或一个服务器同时与多个终端通信的情况下，接收终端可能是异构的(QoS 能力不同)，由接收端各自申明自己要求的服务质量启动资源预留，比由发送端(服务器)向网络提出 QoS 要求更为合理。但是接收端启动也存在问题，因为接收端并不直接了解数据包所经过的路径。为了解决这个问题，发送端在会话开始时要发送一个称之为“路径”(Path)的消息，该消息途径各路由器到达收端(见图 7-18)。Path 消息中不仅包含了有关发端的信息，如发端 IP 地址、端口号、发送的数据格式以及以令牌桶形式定义的业务流特性，还包含了由沿途路由器所更新的“路径状态”信息，如前一跳路由器的 IP 地址，以及让接收端了解端到端服务情况的信息。接收端收到 Path 消息后，沿原路径返回一个“预留”(Reserve)消息，其中以“**流说明符**”给出所请求的 QoS，如带宽、延时或丢包率；以“**过滤器说明符**”表明哪些数据包享受这一 QoS，即预留的资源是由所有发送者(会议情况下)共享的，还是其中某个(或几个)发送者独享的。网络在回传 Reserve 消息时，沿途路由器根据自己的资源情况可以接受、或拒绝这个请求。如果拒绝，则返回一个错误消息给接收端，呼叫被中止；如果接受，则为该业务流分配带宽和缓存空间。Path 和 Reserve 消息可以由协议号为 46 的 IP 包传送，也可以打成 UDP 包传送。

值得注意，RSVP 预留资源是从发端到收端单方向的，双向预留需要两方终端分别启动预留机制。另外需要注意的是，路由器预留的资源只维持一段时间，发端和收端必须周期性地发送 Path

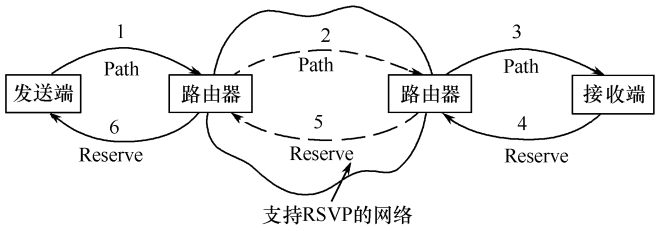


图 7-18　资源预留协议

和 Reserve 消息来保持预留状态，否则资源将被释放。这称为**软状态**。软状态有利于多媒体会议成员动态进入和退出情况下资源的有效利用。

综合服务模型由如图 7-19 所示的 4 个部分来具体实现：

(1) RSVP。RSVP 不负责路由，它实际上是一个信令协议，即要求负荷受控或可保障服务的发送端和接收端必须先通过 RSVP 建立起路径并预留好资源以后，才能开始数据的正式传送。在会话结束后，则用“撤销”(Tear-down)消息撤销连接。

(2) 监管控制和接纳控制。监管控制负责检查用户是否具有预留资源的权力(还可能包括认证、计费和接入控制)；接纳控制负责监测系统资源和决定节点是否有能力接受一个新的流。在这两项检测通过后，RSVP 守护才将有关的控制信息送给包分类器和调度器，完成资源预留。

(3) 分类器。当路由器接收到一个数据包后，分类器根据过滤器说明符将包分类，并按服务类型将包送到不同的队列中去。

(4) 包调度器。调度器按包的 QoS 要求(流说明符)进行调度。

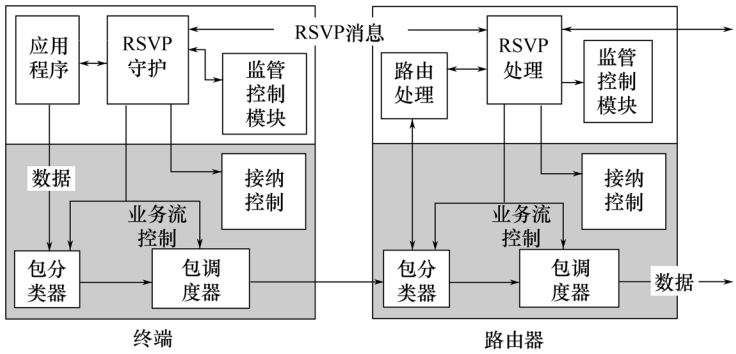


图 7-19　综合服务模型的实现

要获得可保障的服务，整个链路上的路由器都必须是支持综合服务的(即包括以上 4 部分)。如果链路上的路由器有的支持综合服务，有的不支持，则只能获得负荷受控服务。

7.8.2　区别服务

由于综合服务模型和 RSVP 实现起来比较复杂，IETF 在 RFC 2475 中提出另一种 QoS 保障机制，称为**区别服务模型 DS**(Differentiated Services，**Diffserv**)。与 DS 有关的建议见 RFC 2427～74 和 2597～98。

1. 服务类型

在 IP v4 的数据包的包头中有一个字节的“服务类型”域，该字节被分为如图 7-20 所示的 5 个子域，其中 3 个比特表示优先级，D、T、R 3 位表示所

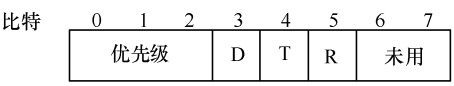


图 7-20　IP 数据包头中服务类型域的子域结构

希望的传输类型，D 代表低延时，T 代表高吞吐率，R 代表高可靠性(低丢失率)。在 IPv4 中上述要求都是用户指定的，不带有强制性，在多数情况下网络往往忽视这些请求。

DS 利用 IPv4 服务类型域(或 IP v6 业务等级域)来区别服务的类型，在 DS 中，它们被称为 **DS 域**。通过 DS 域的前 6 个比特可以定义丰富的服务类型，以及一套对于不同类型的数据包如何进行传递的方式。这 6 个比特称为**区别服务码点** DSCP(Diffserv Codepoint)，而这些传递方式则称为**每段性能** PHBs(Per-Hop Behaviors)。根据 DS 域标识的不同类型，将数据包以不同的方式传递，这便是区别服务(DS)这个名称的由来。

DS 域中可定义的服务有：要求低延时、低抖动的最高服务(Premium Service)，要求比“尽力而为”服务有更高可靠性的确保服务(Assured Service)和具有金、银、铜三种质量的奥林匹克服务(Olympic Service)。

最高服务采用快速传递方式，称为 **EF PHB**(Expedited Forwarding PHB)，提供穿越 DS 网络的低延时、低抖动、低丢包率、保证带宽的端到端的服务。对于用户而言，相当于一条虚拟租用线路，也类似于 ATM 中的 CBR 服务。最高服务的推荐码点为 101110，标有此码点的数据包在路由器中被送往特定的队列。同时，路由器为 EF PHB 配置一个最小输出速率，只要输入速率小于最小输出速率，延时和带宽要求就能得到保障。

其余 4 个类别的服务采用有保障的传递方式，称为 **AF PHB**(Assured Forwarding PHB)。类别不同提供质量不同的服务，例如金牌服务(AF3)的数据包比银牌(AF2)的包所途经的负荷要轻一些。在同一类服务中，又可进一步划分为三个丢包级别，例如在银牌服务中，码点为 AF21＝010010 的包比 AF22＝010100 的包丢失的概率要小一些。

2. 区别服务模型

用户要想获得区别服务必须先和因特网服务提供商(ISP)协商取得**服务水平协定 SLA**(Service Level Agreement)，该协定规定了提供给用户的服务等级和每个等级所允许的流量。SLA 可以是静态的，也可以是动态的。静态 SLA 是用户和 ISP 协商好的、在一定期限(如 1 个月)内有效的协定，在规定期间内用户可以随时享受区别服务；而动态 SLA 是用户在需要区别服务时，通过信令协议(例如 RSVP)建立起来的。当 SLA 建立起来之后，用户对每一个数据包的 DS 域进行标识。在 ISP 网络的入口处，接入(或边界)路由器根据 SLA 对数据包进行分类和不同的处理(见图 7-21)。

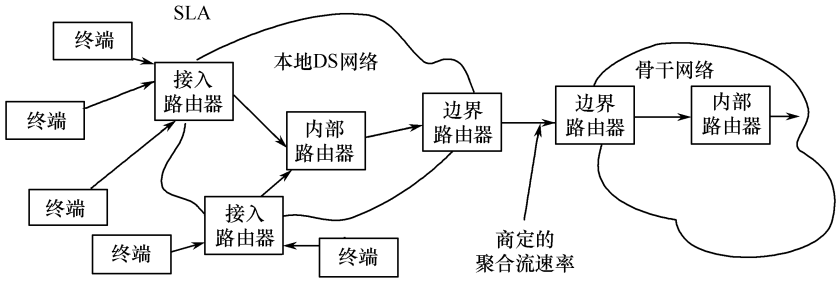


图 7-21　DS 网络结构示例

接入路由器按服务类别将每个输入包流(又称为**微流**)的数据包分类[见图 7-22(a)]。包被分类后，由图中测量器测量微流是否遵从协商好的业务流特性，如峰值和平均速率、突发度等，并将微流遵从约定的情况通知标记器，决定是否将一部分包改为较低的服务等级；同时通知丢包器决定包是否丢弃。流成形和丢包操作使进入本地 DS 网络和进入核心网的流能够符合约定的特性。在核心(内部)路由器中(见图 7-22(b))，操作只针对聚合流进行。具有相同 DS 码点的微流的聚合称为**行为聚合流 BA**(Behavior Aggregate)。不同 BA 的包进入不同的队列，并使用不同的调度和丢包策略(PHB)处理。

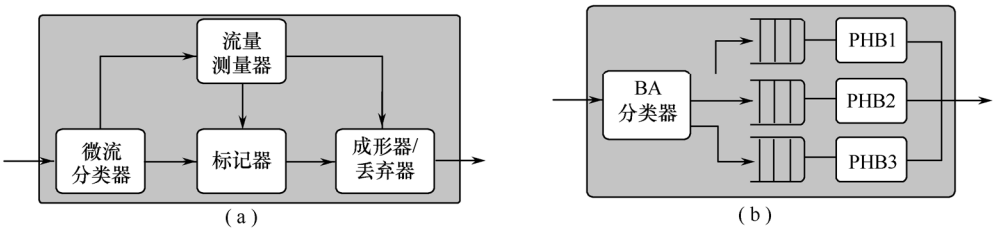


图 7-22　DS 模型的实现(a)边界路由器;(b)骨干路由器

如果链路上的路由器有的支持 DS,有的不支持,不支持 DS 的路由器忽视数据包中 DS 域的标识,只给予“尽力而为”的服务。但是由于支持 DS 的路由器对要求特殊服务的包会给予应有的处理,因此从整体性能来看,要求特殊服务的包还是得到了比“尽力而为”更好的服务。

3. 区别服务与综合服务的比较

区别服务的实现要比综合服务简单,因为二者之间有如下的根本性的区别:

(1) 在综合服务中,用户对每一条流提出不同的 QoS 要求,而 DS 只提供有限的服务类别,因此在 DS 中路由器只需记录不同类型服务的状态信息,而不像在综合服务中那样,需要花费大量的存储空间和处理能力来记录每一条流的状态信息;

(2) 在 DS 中,对报文进行复杂的分类、标识和处理过程均在用户和网络的交界处进行,核心路由器只对聚合流进行操作,因此较为简单,有利于在网络核心部分实现包的快速传递;而用户速率一般较慢,允许与之相连的边界路由器花费较多的时间来进行报文分类和处理。在综合服务中则不是这样,所有的路由器都必须支持图 7-19 所示的 4 种功能,因此对路由器的要求是很高的。

表 7-1 给出了两种模型的概要比较。区分服务的缺点是它不能实现端到端的确定的 QoS 保障,它所提供的只是一种 CoS 的相对的服务质量。

表 7-1　综合服务与区分服务比较

	Intserv	Diffserv
包处理	特定流	服务类别
传递服务	端到端	段到段
资源分配	信令启动	静态或动态 SLA
路由器	复杂	核心路由器简单
规模伸缩性	差(用户增加,消耗的资源大大增加)	好

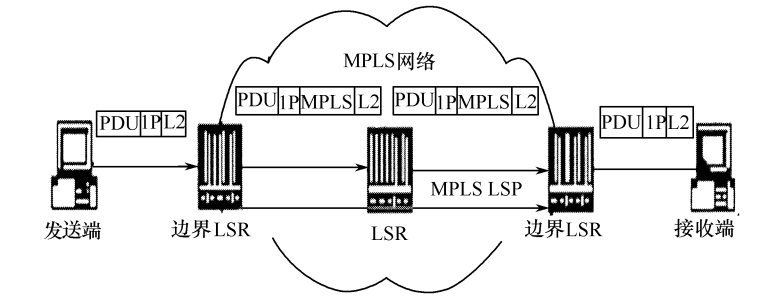
7.8.3　多协议标记交换 MPLS

多协议标记交换 MPLS(Multiprotocol Label Switching)是 IETF 提出的又一种 QoS 保障机制(RFC 3031～32)。在 ISO 的七层模型中,它位于第二和第三层之间。

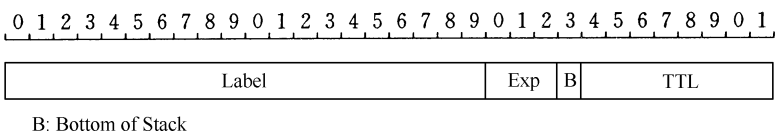
1. MPLS 的基本工作过程

图 7-23(a)给出一个 MPLS 网络示意图。发送端的用户数据包在 MPLS 网络的入口(边界)路由器中进行分类、并在 IP 包头和第二层(链路层)包头之间加入相应的 MPLS 包头,然后进入 MPLS 网。MPLS 包头如图 7-23(b)所示。包头的前 20b 为一个**标记**(Label),接着是 3b Exp(Experimant)域、1b 标记堆栈指示和一个 8b 的生存期(TTL)域。标记用来标识一条虚电路,支持 MPLS 的路由器、或称为**标记交换路由器 LSR**(Label Switched Router)根据标记对包进行转发和处理。Exp 域常常用来表示服务等级(CoS)。在 MPLS 网络中以相同方式(沿相同路由、相同服务等级)传送的数据包称为**转发等价类 FEC**(Forwarding Equivalence Class)。一个特定的数据包属于哪个等价类是在入口路由器中决定的。在 MPLS 网络的出口(边界)路由器中,数据包的 MPLS

头被去掉，然后进入另一种网络。从入口到出口的单向逻辑通道称为**标记交换路径 LSP**（Label Switched Path），它相当于穿过 MPLS 网络的管道。从不同源来的、具有相同 FEC 的数据流可以聚合共享一个 LSP。



(a)



(b)

图 7-23　MPLS 网络示意图

MPLS 在传输数据之前，需要根据数据包的目的地址通过路由协议寻径，但一旦从入口到出口路由器的 LSP 建立并获得了对应的标记（通过 MPLS 的信令和控制协议——**标记分发协议（LDP）**来完成）之后，后续包的传送则只根据标记、而不再关心原来的 IP 地址。

2. MPLS 的 QoS 功能

如前所述，MPLS 是面向连接的，且可以对不同服务等级的包提供不同的服务。由于包的 FEC 分类是在边界路由器上进行的，因此 MPLS 很容易与 Diffserv 这样的 QoS 机制相融合。E-LSP（Exp-infered-LSP）和 L-LSP（Label-infered-LSP）[4] 就是 MPLS 与 Diffserv 结合的两种方式。

在传统的 IP 路由协议中，路径（例如最短路径）完全由目的 IP 地址决定。在 MPLS 中，允许具有同样目的地址的业务流遵循不同的预定路径传输，也就是说，既可以从最短路径，也可以从显式（Explicit）指定的其他路径传输。后者使 MPLS 成为实现下节将要介绍的流量工程的很好的平台。

3. MPLS 的路由与交换功能

在讨论 RSVP 时，我们强调了 RSVP 不负责路由。但与 Intserv 与 Diffserv 不同，MPLS 不仅是关于 QoS 机制的协议，它更重要的是有关路由和交换的技术。传统的 IP 路由器按照目的 IP 地址来查看和确定下一段的路由，这种靠最长前缀匹配查表的方法一般用软件实现，是妨碍路由器处理速度提高的重要因素。而 MPLS 采用的是类似于 ATM 虚电路的方法，用一个短的固定长度的标记作为路径连接表的索引，因此很容易利用高速的硬件来实现包的交换。也就是说，MPLS 将 ATM 交换的原理由交换机内部扩展到整个网络的范围。

MPLS 的一个重要特点是在节点处控制功能和转发功能的分离，图 7-24 给出了这个概念的图示。在 LSR 中，控制单元负责路由的选择、控制协议 LDP 的执行、标记的分配与发布及标记信息库的形成等；而转发单元则只负责根据标记信息库建立标记转发表和对已标记的包进行简单的转发操作。这种分离的方式允许两个功能模块分别独立地演变和扩展，这给系统的发展提供了极大的灵活性。MPLS 集成了网络第三层的路由功能和第二层的转发（交换）功能，它的控制部分能支持 IP（v4 和 v6）路由或任何其他第三层协议，而转发功能则可以建立在多种第二层技术之上，例如 ATM、帧中继、PPP 和以太网等。这就是在 MPLS 名称中强调“多协议”的由来。

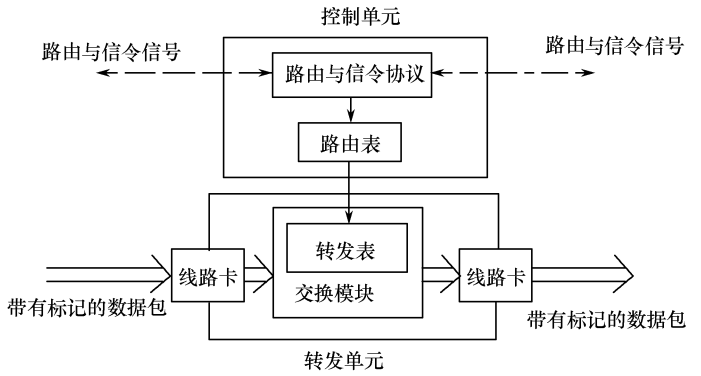


图 7-24　标记交换中控制和转发功能的分离

通用 MPLS(Generalized MPLS,**GMPLS**)是 IETF 和 OIF(Optical Internetworking Form)对 MPLS 扩展的一组协议,其框架由 RFC 3945 定义。GMPLS 利用 MPLS 控制和转发功能分离的特点,着重对控制功能进行了扩展。它通过原来 MPLS 控制部分中的大部分功能以及一些新加入的扩展功能来建立(和撤销)不同类型的连接通路。这使得它不仅支持 ATM、帧中继等分组(包)交换技术,也能够支持时分复用(TDM)交换、λ 交换(即波长交换)和光纤(端口)交换等技术。GMPLS 允许在一个节点中用一个统一的控制模块控制所有这些交换方式,从而建立起不同类型的连接通路。这大大降低了异构传输网络集成时在网络控制和管理方面的成本,也使得 GMPLS 成为下一代核心网最重要的技术之一。

7.8.4　流量工程和基于约束的寻径

综合服务和 DS 在网络负荷较大的情况下,能够保证 QoS 要求较高的数据流的性能下降较小。但是当网络负荷很轻时,"尽力而为"的网络性能也很好,与综合服务和 DS 相差很小,那我们为什么不从避免拥塞的角度来解决 QoS 保障的问题呢? 这就是**流量工程**(Traffic Engineering,TE)[5]的出发点。

引起网络拥塞的原因可能是全部网络资源已经耗尽,即所有的路由器和链路都过负荷;也可能是由于网络负荷的不均衡,例如使用选择最短路径的动态路由协议(RIP,OSPF 和 IS_IS 等[2])往往会造成在两个节点的最短路径上的路由器和链路发生拥塞,而最长路径上的路由器和链路空闲。在前一种情况中,消除拥塞的唯一方法只能是将网络的基础设施升级;而对于后一种情况则存在更有效地利用已有设施提高网络性能的余地。流量工程就是针对这种情况,设法均衡网络流量以避免拥塞的措施,而**基于约束的寻径**(Constraint-based Routing,CR)[6]则是自动实现流量工程的一个重要工具。

基于约束的寻径,顾名思义,是在寻径时考虑到多个约束条件,它由 QoS 寻径演变而来。在 QoS 寻径中,对于给定 QoS 要求的某个业务流(或会聚业务流),路由器选择的是最有可能满足其 QoS 要求的路径。而基于约束的寻径不仅考虑 QoS 要求,而且考虑所选择的路径是否满足网络负荷均衡的要求,因此,在确定一条路径时,其依据不仅是网络的拓扑结构,还包括业务流的 QoS 要求、链路当前可利用的资源,以及其他一些网络管理者制定的有关规定。

为了实现基于约束的寻径,路由器需要不断地将链路的当前状态信息(例如剩余带宽)公布给其他路由器,并根据这些信息计算路径。在计算最佳路径时可以考虑的约束条件有带宽、延时、延时抖动、错误率、路径的段数和费用等。但是约束条件太多,使该最佳化问题变得十分复杂。比较可行和合理的办法是只考虑带宽和路径段数的约束,此时寻径算法比较简单,常见的 Bellman-Ford 算法 或 Dijkstra 算法都可以使用[7]。由于即使在网络拓扑结构不变的情况下,链路状态信息

也随时间不断变化，因此在基于约束的寻径中，路由器的计算量是很大的。

流量工程的一个实例就是它在 MPLS 中的应用。我们知道，在 MPLS 中，在数据传输之前需要通过标记分发协议 LDP 来建立 LSP。基本的 LDP 借助于传统的 IP 路由协议（如 OSPF、BGP 等[2]）进行寻径。引入流量工程的思想，我们则可以在入口 LSR 向出口 LSR 请求标记的时候，同时给出有关的 QoS 参数，中间路由器根据这些参数进行基于约束的寻径和相应的资源预留，从而建立起 LSP。这种支持流量工程的 LDP 称为 **CR-LDP**[8]。利用扩展的 RSVP 信令协议（**RSVP-TE**）[9]代替 LDP 也可以实现 MPLS 中的流量工程，此时原来在收发主机之间的协商扩展到在入口和出口 LSR 之间进行。值得指出，当网络状况发生变化时，CR-LDP 和 RSVP-TE 都允许在不中断业务和不重复进行带宽预留的条件下，让业务流从一条 LSP 快速地切换到另一条流量较小的 LSP 上去。

7.9 无线网络对多媒体信息传输的支持

7.9.1 无线传输的特点

无线连接省去了布线，为组建网络带来很大的灵活性和方便性；同时允许用户终端是移动的，这包括移动到新的地点再接入和在移动的过程中持续进行通信两种情况。与有线（光缆、铜缆等）网络相比，无线传输存在着其特殊的问题。

（1）频带利用

无线电波是指频率从 100 kHz 到 100 GHz 的电磁波，其中按照波长的长短可分为长波、中波、短波、超短波、微波、红外等频段。波长越短，天线的尺寸越小。无线信道具有广播性，一个发射机发送，在其电波覆盖范围内的多个接收机可以同时接收。由于在同一个区域内两个发射机使用同一频率进行发送会形成相互干扰，因此各个国家对频率的使用都需要经过授权。这也说明频率资源的宝贵。频带利用率是无线通信系统设计的一个重要指标。

（2）信号衰减和噪声干扰

无线信号的能量随着传播距离 D 的加长而衰减。在真空中能量按 D^{-2} 衰减，在实际环境中则有可能按 $D^{-3.5}$，甚至 D^{-5} 衰减。信号衰减到一定程度，将不能正确地接收。同时，无线电波在开放的空间中传播容易受到外界噪声和干扰的影响，这导致无线信道的误码率比有线信道高得多，例如光缆传输的误码率通常在 $10^{-12}\sim10^{-9}$，而无线信道中则达到 $10^{-4}\sim10^{-3}$，严重时甚至可到 10^{-2}。对于使用无需进行频率申请的频段如工业、科学和医学应用的频段的网络，例如无线局域网，不仅可能受到相邻同类网络的干扰，还可能受到其他设备，例如日光灯、微波炉等的干扰。

（3）多径效应

在发射机电波覆盖范围内的障碍物（如家具、建筑和山川等）都是无线电波的散射体，障碍物产生与入射波同频的、幅度畸变而相位依赖于入射角的反射波。此时用户终端除了接收到一个从发送端来的直达波外，还接收到多个不同角度到达的反射波，在不同时间从不同角度到达的反射波可能具有不同的相位，它们之和构成了在空间和时间上不断变化的干扰场，这称为**多径效应**。由于干扰场的影响，直达信号在很短距离上的衰落可能达到 20 dB。同时，由于各种反射波到达接收端的路径（传输延时）不同，发送一个窄脉冲，接收信号可能会是一个脉冲串，这种现象称为**时间弥散**，它会引起码间干扰。

（4）多普勒频移

当在通信过程中接收终端发生移动时，所有频率均产生与移动角速度成正比的频率偏移，称为**多普勒频移**。因此发送一个单音信号，可能接收到一个非零带宽频谱的信号。

由于无线信道的上述特点，使得物理层技术成为无线网络设计中的关键问题。

7.9.2 无线局域网

无线局域网,简称 **WLAN**(Wireless LAN),是通过无线介质(无线电波或红外波)连接的室内或园区计算机网络。IEEE 802.11 系列的 WLAN 是无线局域网的典型代表。WLAN 工业界的 Wi-Fi 联盟(Wireless fidelity Alliance)制定的 **Wi-Fi** 标准与 IEEE 802.11 兼容。

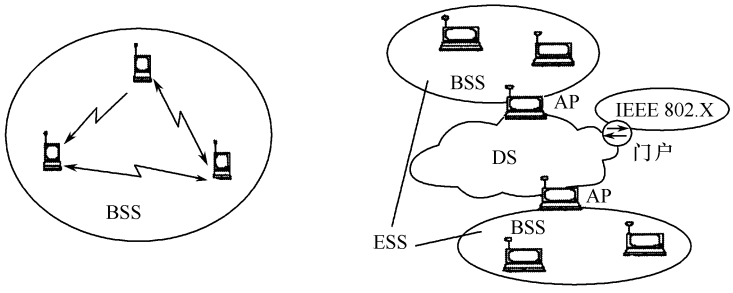
802.11 系列是关于 MAC 层和物理层的标准。它定义了若干不同的物理层。在各物理层之上使用同样的 MAC 子层;在 MAC 子层之上,提供以太网类型的服务。在本节中我们重点介绍 MAC 子层,因为它包含研究多媒体信息传输所关心的对 QoS 的支持。

1. 无线局域网的构成

WLAN 的基本结构单元称为**基本服务集** BSS(Basic Service Set)。一个 BSS 是一组站点,它们由给定的 MAC 协议来解决对共享介质的接入。由一个 BSS 所覆盖的地域称为基本服务区域 BSA(Basic Service Area),其最大直径为 100m。WLAN 可以两种方式组网:自组织(Ad hoc)网络和基础设施(Infrastructure)网络。单个的 BSS 即为一个**自组织网络**[见图 7-25(a)],这样的网络可以随时随地组成和拆除。一组 BSS 各自通过一个**接入点 AP**(Access Point)连接到**分配系统 DS**(Distribution System)中则构成扩展服务集 ESS(Extended Service Set)。在这里 BSS 类似于移动网中的蜂窝,AP 则类似于基站(见 7.9.3 节)。ESS 可以通过称之为门户(Portal)的设备连接到网关上,实现与有线网络的互联。这组 BSS、DS 和门户就构成了**基础设施网络**[见图 7-25(b)]。

在一个 ESS 中,DS 的功能是:(1)在 AP 之间和 AP 与门户之间传递 MAC 层的服务数据单元(SDU);(2)当 MAC SDU 具有多播或广播地址时,或当发送站选择使用分配服务时,负责同一 BSS 中各站之间的 MAC SDU 传送。总之,在 ESS 中,DS 使站点主机 MAC 层以上的协议层感觉它们好像是处在同一个 BSS 中一样。DS 可以通过无线、或者有线网络来实现。

如果一个站点想要加入图 7-25(b)的网络,它首先需要选择一个 AP 与之建立联系,然后才能通过该 AP 收、发数据。当它欲离开网络时,需要撤销与该 AP 的联系。802.11 标准还定义了再建连(Reassociation)服务,允许站点将与一个 AP 已经建立的联系转移到另一个 AP。



(a) Ad hoc网络　　(b) 基础设施网络

图 7-25　无线局域网的构成

IEEE 820.11 已经定义了几个物理层标准,表 7-2 给出了它们的一些主要参数。

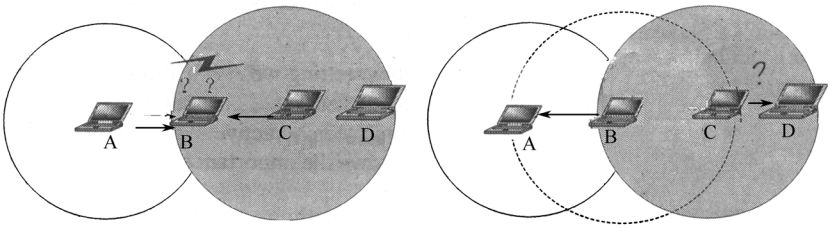
表 7-2　802.11 物理层标准

	802.11	802.11b	802.11a	802.11g	802.11n
最高数据率	2 Mb/s	11 Mb/s	54 Mb/s	54 Mb/s	600 Mb/s
使用频带	2.4 GHz	2.4 GHz	5 GHz	2.4 GHz	2.4 G/5 GHz
信道带宽	20 MHz	20 MHz	20 MHz	20 MHz	20/40 MHz
空间流数目	1	1	1	1	1—4
兼 容 性			不兼容	与 802.11b 兼容	与 a,b,g 兼容

802.11n 中采用了更宽的信道带宽和**多输入多输出** MIMO(Multiple Input Multiple Output)技术。MIMO 使用多天线系统,通过每个天线发送的独立数据流(空间流)经过空分复用后,其总数据传输速率相比于采用单天线系统的先前的 802.11 标准有很大的提升。同时,由于智能天线的使用,802.11n 的覆盖范围扩大到几平方公里,这使得 WLAN 的移动性有了很大提高。

2. 分布式协调功能(DCF)

虽然 WLAN 在高层提供以太网类型的服务,而且和以太网一样是一个广播型的网络,但它却不能采用以太网的 MAC 协议——CSMA-CD。原因是:(1)同一站点的发送信号功率通常远大于接收信号功率,因此不能像有线网那样通过检测"总线"上的异常信号电平来发现"碰撞"。(2)在有些情况下,例如图 7-26(a)所示,A、C 两站同时欲向中间的 B 站发送信号,而 AC 之间的距离又大于各自的电波覆盖范围(如图中圆圈示),即二者相互侦听不到对方的发送信号,因此二者可能同时向 B 传送信号,导致在 B 站信号的碰撞。这称为**隐藏站问题**。在另外一些情况下,如图 7-26(b)所示,B 正在向 A 发送数据(B 不接收),而与 B 相距较近的 C 欲向 D 发送数据。由于 C 在 B 的覆盖范围之内,它侦听到 B 向 A 传送的信号而误认为通向 D 的无线信道已被占用,事实上 D 在 B 的覆盖范围之外,C 是可以向 D 传送信号的。这称为**暴露站问题**。暴露站问题不会引起"碰撞",但降低了频带资源的利用率。



(a) 隐藏站问题　　(b) 暴露站问题

图 7-26　相邻站之间的侦听

由于不能通过侦听来发现碰撞,因此在 WLAN 中使用肯定确认(ACK)的方法来通知发端发送是否成功。如果收不到 ACK,说明碰撞(或噪声)致使传送的数据丢失,需要重新发送。为了减少碰撞引起的带宽损失,WLAN 的 MAC 协议注重碰撞的回避,因而称为 **CSMA-CA**(Carrier Sensing Multiple Access with Collision Avoidance)。如果一个站侦听到在规定长度的时间段内,该时间段称为**帧间距离 IFS**(Inter-frame space),或者更长的时间内信道一直是空闲的,从原理上讲,它可以开始发送数据。但为了防止侦听到 IFS 信道空闲的多个站同时发送数据而形成碰撞,协议要求每个站还要继续侦听一段时间,这称为进入**退避**(Back off)**状态**。继续侦听的这段时间称为**竞争窗**,各站竞争窗的大小是随机选择的。如果在竞争窗内信道一直空闲,站点在窗结束时进行发送。由于各站竞争窗的大小不同,因此大大降低了信号发生碰撞的概率。如果此发送不成功(无 ACK),则该站须重新进入一个新的退避状态,且竞争窗加大一倍。以上过程称为**物理载波侦听**。

除了物理侦听外,WLAN 的 MAC 层还需要进行**虚拟载波侦听**。这实际上是一套握手机制:在传送数据前,发送端[如图 7-26(a)中 A]先向收端(如图中 B)发送一个 RTS(Request-to-send)帧;如果接收端 B 允许接收,则返回一个 CTS(Clear-to-send)帧。侦听到 CTS 的其他站(在 B 的覆盖范围而未必在 A 的覆盖范围之内)必须等待一段时间以让 A 完成发送,这解决了隐藏站问题。而侦听到 RTS 帧而听不到 CTS 帧的其他站,则可以尝试进行另外的发送(发送另外的 RTS),因为它们的信号可能不会对 AB 之间的通信造成影响,这解决了暴露站问题。

发送站或接收站在自己 RTS 或 CTS 帧的帧头中给出接下来的数据传输(包括 ACK)所需要的时间,即其他站需要等待的时间。收到 RTS 和/或 CTS 的其他站根据此值将自己的一个称为**网**

络分配矢量(Network Allocation Vector,NAV)的时间计数器置位,等到这段时间结束后再对信道重新进行侦听。收到 RTS 的站也要将 NAV 置位,是为了防止在 ACK 期间发送区域内的碰撞,这种碰撞可能由在接收站覆盖范围之外的站引起。

CSMA 和 RTS/CTS 握手机制构成了 WLAN 的基本接入方式,称为**分布式协调功能 DCF** (Distributed Coordination Function)。它既可以用于自组织网络,也可以用于基础设施网络。站点发送每一个帧都需要通过物理和虚拟载波双重侦听,才能取得介质的使用权。图 7-27给出一个站 2 欲向站 1 和站 4 欲向站 3 进行发送的示例。图中 DIFS 表示采用 DCF 时的帧间距离,SIFS 表示较短的 IFS(short IFS),即 SIFS<DIFS。在发送 CTS、数据和 ACK 之前只等待 SIFS,这说明这些帧具有较高的优先级。图中站 2 和站 4 的竞争窗分别为 7 个和 9 个时隙,站 2 先结束退避状态,因而获得了介质使用权。值得注意的是,站 4 在竞争窗内侦听到介质被占用,它将自己的竞争窗计数暂停,并保留剩余的窗大小(两个时隙)。在 NAV 结束后,由于剩余窗较小,站 4 获得了介质的使用权。这种方法有利于保证各站接入的公平性。站 6 收不到站 2 的 RTS,但能收到站 1 的 CTS,它根据 CTS 置位自己的 NAV。

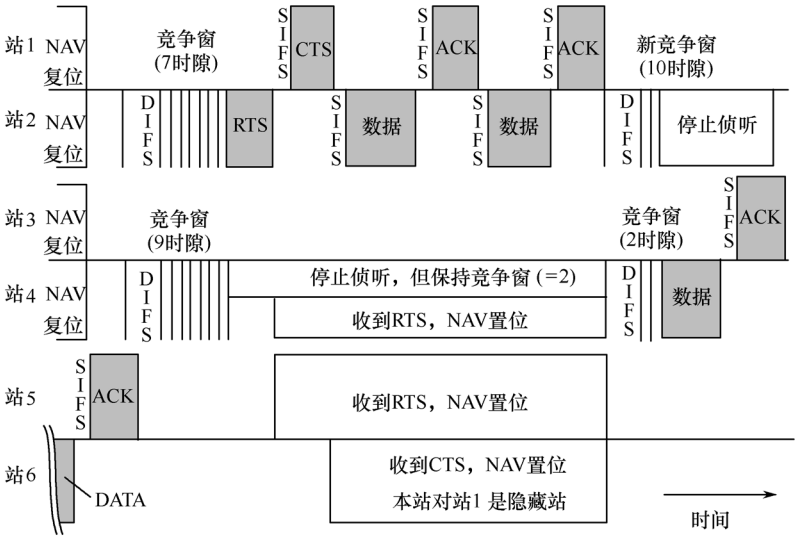


图 7-27　DCT 时序

在使用 WLAN 时,有两个参数值得注意,它们分别是 RTS-Threshold 和 Fragmentation-Threshold。RTS/CTS 机制防止了隐藏站引起的数据帧碰撞,但是两个欲发送的站同时产生的 RTS 帧还有可能发生碰撞。由于 RTS/CTS 帧比数据帧短,例如 RTS 和 CTS 分别为 20 B 和 14 B,数据帧可能为 2 300 B,因此 RTS 碰撞比数据帧碰撞损失的带宽要小得多。但在网络负载轻(碰撞概率小)时,RTS/CTS 机制在数据传输中引入了不必要的延时。为了适应不同的应用环境,标准允许选择使用或不使用 RTS/CTS,或者选择一个适当的阈值(RTS-Threshold),当数据帧大于此阈值时使用 RTS/CTS。

在 8.2.3 节中我们将会看到在误码率较高的信道(例如无线信道)上不宜传送较长的包,上层较长的数据包到达 MAC 层后,常常需要分段封装成若干个 MAC 帧来传输。标准允许选择一个适当的阈值(Fragmentation-Threshold),当包长大于此值时,则进行分段。

3. 点协调功能(PCF)

从前面的讨论可以看出,DCF 提供的是尽力而为的服务,网络上的各站点公平地竞争介质的使用权;而下面将要介绍的 PCF、EDCF 和 HCF 则提供不同程度的 QoS 服务。PCF、EDCF 和

HCF 建立在 DCF 之上(见图 7-28),它们与 DCF 并存,并只能在基础设施网络上应用。

点协调功能 PCF(Point Coordination Function)支持面向连接的非竞争性服务。在一个 BSS 中,位于 AP 内的一个**点协调器** PC(Point Coordinator)负责控制各站的介质使用权,这有利于对实时媒体流传输带宽的保障。

PC 周期性地发送一个**信标**(Beacon)帧,标志一个非竞争重复周期的开始(见图 7-29)。发送信标帧所等待的时间 PIFS 要小于 DIFS(但大于 SIFS),这意味着 PC 比普通站有更高的发送优先级。非竞争重复周期的长度是一个可调整的参数,它通过信标帧告知各个站。整个周期的一部分由非竞争业务使用,余下的为竞争(DCF)业务使用。在周期起始处,所有站将自己的 NAV 计数器置到最大非竞争时间(预设参数),在此期间所有站都处于等待状态。PC 在发送信标帧之后发送一个帧,它可以是非竞争授予(CF-Poll)帧、数据帧或数据+CF-Poll,其中 CF-Poll 指定哪一个站可以发送数据。被指定发送的站返回一个帧,它可以是非竞争应答(CF-Ack)或者数据+CF-Ack。PC 在收到应答帧后再发送一个混合帧,其中 CF-Poll 指定下一个可以发送的站,而 CF-Ack 是对前一帧的应答。当 PC 发送一个 CF-End 帧时,表示非竞争期的结束,所有站将自己的 NAV 置零,从而进入 DCF 竞争期。

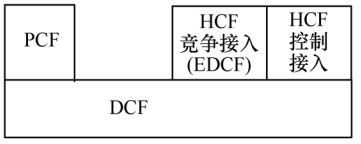


图 7-28　820.11 MAC 层结构

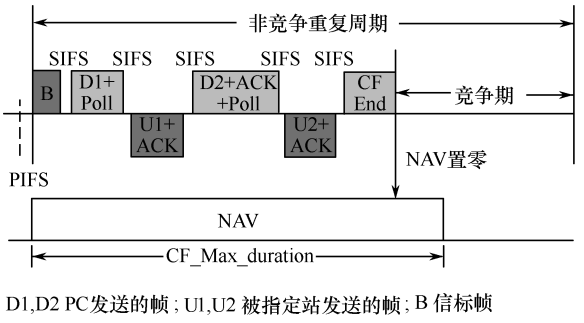


图 7-29　PCF 帧传输

PCF 提供了一定程度的 QoS 保障,但还存在不少问题。首先,在 PC 需要发送信标帧启动下一个非竞争重复周期时,上一个周期内的竞争业务可能尚未结束传输,PC 需要等到介质空闲才能进行下一个信标的发送。每一个周期信标的延时可能不相同,造成各周期的非竞争期传输时间的抖动,这对实时媒体的传输是不利的。其次,在非竞争期间,被指定发送的站所占用的发送时间(帧大小)也是不可预知的。当站在通信期间漫游产生物理层切换(速率变化)时,其发送时间 PC 更无法控制,这同样对时间敏感媒体的传输不利。另外,PC 在非竞争期采取轮巡策略轮流授予它的授予站列表中各站发送权,并不管它们当前是否有待发送的数据。显然,这造成一定的带宽浪费。同时,低延时的应用希望非竞争重复周期较短,而长的周期有利于带宽的有效利用。PCF 的非竞争重复周期不能动态地调整,难以满足不同应用情况下的折中。值得指出,在 PCF 尚未真正进入实际应用时,更好的支持 QoS 的机制——802.11e 已经出现,因此可以预见 PCF 将不会被广泛使用。

4. 增强型 DCF(EDCF)

802.11e 与表 7-2 列出的 802.11a、b 和 g 不同,后者是关于物理层的标准,而 802.11e 是关于 MAC 层的标准。802.11e 规定了两种进一步支持 QoS 的接入协调功能:EDCF 和 HCF。

增强型分布式协调功能 EDCF(Enhanced DCF)支持最多 8 种服务等级(优先级)。如图 7-30 所示,具有不同优先级的包进入 MAC 层后分别映射到 8 个**接入类别**(Access Category,AC)的队列中。这些 AC 共同竞争一个**传输机会** TXOP(Transmission Opportunity),并各自独立地进行退避操作,这称为**虚拟 DCF**,就好像一个站包含了 8 个虚拟站一样。每一个 AC 有自己可设置的一组参数,如帧间

距 AIFS(Arbitration IFS)、最小和最大竞争窗 CWmin 和 CWmax、持续因子 PF(Persistence factor)等，其中 AIFS≥DIFS，PF 决定在发送不成功时新竞争窗的大小(在 DCF 中，新竞争窗简单地扩大为旧窗的 2 倍)。PC 可以根据当前网络状态动态地调整这些参数。虽然帧间距和竞争窗越小，其对应的 AC 优先级越高，由接入而引入的延时越小，但竞争窗减小，碰撞的可能性加大。当不止一个 AC 同时结束自己的退避状态时，调度器将 TXOP 给予优先级最高的 AC，其他 AC 进入新的退避状态，从而避免了本站内部的虚拟碰撞。

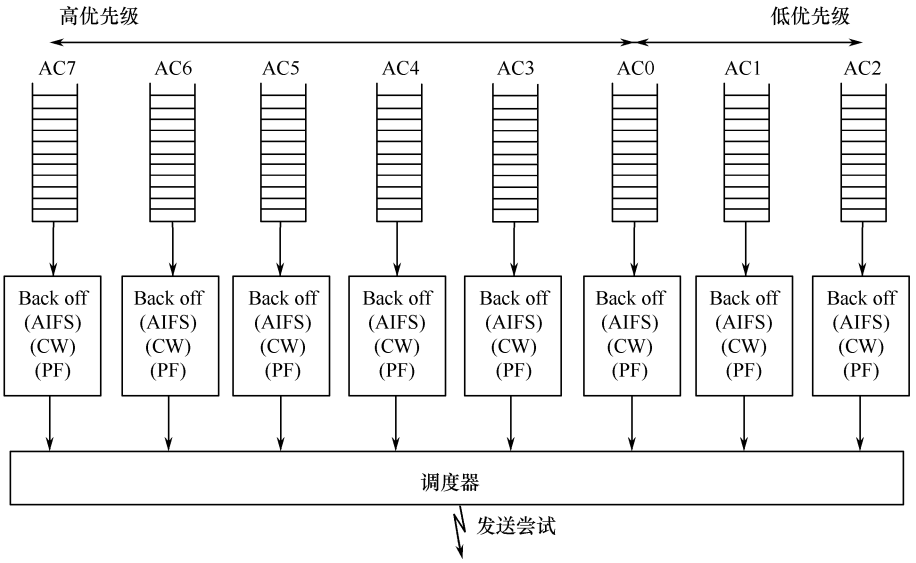


图 7-30　EDCF 实现模型

5. 混合协调功能(HCF)

混合协调功能 HCF(Hybrid Coordination Function)使用一个能感知 QoS 的点协调器，称为**混合协调器 HC**，来控制介质的接入。与 PCF 中的 PC 类似，HC 通过周期性地发送信标帧启动非竞争重复周期。整个重复周期分为非竞争期和竞争期两部分。在非竞争期内，由 HC 按照划分优先级业务或面向连接业务的 QoS 要求，通过 QoS CF-poll 帧向站授予一定长度的 TXOP。TXOP 的长短决定了该站可以发送数据的多少(可以多于一帧)，这有效地提高了非竞争接入时的信道利用率。同时 HCF 提供了一个称之为**控制竞争**(Controlled Contention)的机制让 HC 了解各个站的 QoS 需求。控制竞争期由 HC 发起，在此期间内，各站向 HC 发送称为业务规范 TSPEC(Traffic Specification)的信令帧，告知 HC 自己期望的 QoS 参数(TXOP 大小)。在非竞争期 HC 则根据各站的报告进行调度，以保障它们的 QoS。TSPEC 的使用也避免了 HC 将发送权授予那些没有数据等待发送的站。

在重复周期的竞争期内，各站按照 EDCF 规则竞争介质的使用权。但是 HCF 的一个重要的特征是，在竞争期间 HC 也可以进行 TXOP 的授予。HC 不经退避只等待 PIFS 空闲(比一般站和 EDCF 有更高的优先级)就可直接发送 QoS CF-poll，授予特定站传输机会。EDCF 和点协调功能的混用，这是混合协调这个名称的由来。

在 HCF 中，HC 清楚地知道它所给出的每一个 TXOP 的长度，这有利于它进行 QoS 调度。同时 HCF 规定如果某个站不能在它的 TXOP 期间、或者不能在下一个信标帧期望到来的时刻之前，完成完整的传输过程(包括 ACK)，则该站不能进行发送，这就避免了下一个重复周期的延时问题。

7.9.3 蜂窝移动通信网

计算机和通信技术的发展使得任何人在任何时间和任何地方以低成本互相进行通信的理想距离现实越来越接近，无线通信则由于能够提供方便的个人通信服务而逐渐成为其中的核心技术之一。使用无线电波作为传输介质的传输网络都称为无线网络。按照系统使用的通信体制和技术，无线网络可以划分成蜂窝移动通信、寻呼移动通信、集群移动通信、卫星通信、微波传输、无线局域网、无线城域网和无线个域网等。在本节中我们关心的是，在终端移动过程中能够保持通信的蜂窝移动通信系统。

1. 蜂窝的概念

蜂窝移动通信系统起源于移动电话业务。早期的移动电话网由一个大功率的基站和移动终端组成。基站的覆盖范围可达 50 km。移动终端与基站通过全双工无线连接进行通信，基站通过有线连接接入到骨干网中。终端在通信过程中不能离开基站的电波覆盖范围，即没有漫游和越区切换的功能。

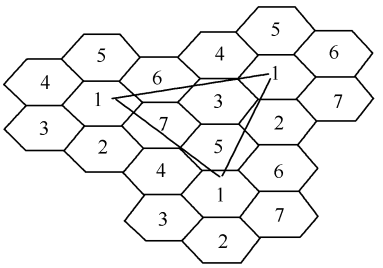


图 7-31　蜂窝的概念

蜂窝概念的提出可以说是移动通信的一次革命。它的基本思想是，试图用多个小功率发射机（小覆盖区）来代替一个大功率发射机（大覆盖范围）。如图 7-31 所示，每个小覆盖区分配一组信道，对应于使用一组无线资源（例如频率）。相邻小区使用不同的无线资源（如图中标号所示），使之相互不产生干扰，相距较远的小区可以重复使用相同的无线资源，这就形成了无线资源的**空间复用**，从而使系统容量大为提高。

使用不同无线资源的 N 个相邻小区构成一个簇，在图 7-31 中，$N=7$，我们称 N 为**重用系数**。N 越大，使用相同资源的小区距离越远，相互干扰越小，但分配给一个小区使用的资源（总资源的 $1/N$）越少。

在蜂窝小区中，移动终端之间不能直接互通，需要通过基站转接。终端向基站的发送称为**上行**线路，反之，称为**下行**线路。终端与基站之间的接口称为无线接口，也称为**空中接口**。基站除空中接口外，还有一个与骨干网连接的接口。

在蜂窝的概念提出来之后，由于系统的覆盖区内有多个小区而用户终端又可以任意移动，这就带来了两个问题：第一，系统如何能够确定用户当前的位置；第二，通信过程中，移动终端从一个小区进入另一个小区，提供服务的基站发生变化，如何保持通信不中断。这两个问题合起来统称为**移动性管理**问题。

此外，蜂窝系统中多个用户之间相互通信还涉及交换的问题；它们与蜂窝系统外的用户进行通信涉及与固定通信网的互通问题。因此，蜂窝系统除基站外，还有基站控制器、移动交换中心、与其他网络互通的节点（网关）、一些用于位置和身份管理的数据库，以及完成鉴权、认证功能的节点等。这些设备合起来构成了蜂窝移动系统的**基础设施**，或称为蜂窝系统的**骨干网**。

2. 多址接入

同一小区内的众多用户终端如何共同使用分配给该小区的一组无线资源，称为**多址接入**问题。WLAN 采用争用信道的 MAC 协议来解决这个问题，而在蜂窝网中采用的方法则是：将资源划分成子信道，每一个子信道分配给一个用户终端使用。根据信道划分方法的不同，有以下三种接入方式。

（1）频分多址（FDMA）

在 FDMA 中，信道可利用的总带宽被分成 M 个互不重叠的子带，每一个终端可以利用分配给

它的一个子带连续地传送信息。因此，FDMA 比较适合于面向连接的流式业务的传送；对于突发式的业务，其资源利用率较低。假设信道支持的总数据率为 R bits/s，在忽略各子带之间的保护频带间隔的情况下，每个终端可传输的最大数据率为 R/M bits/s。

(2) 时分多址(TDMA)

在 TDMA 中，各个终端轮流使用整个信道。将信道的使用时间划分成周期，在每一个 TDMA 周期中又划分 M 个时隙。一个用户终端可以在所分配的时隙中周期性地传送信息。当一个用户的数据率较大时，可以分配给它多个时隙，在这一点上，TDMA 比 FDMA 更为灵活。但在 TDMA 中，在每个时隙之前要加入前导(Preamble)信号，以便于接收机与所接收信号的同步，这增加了一些开销。假设信道支持的总数据率为 R bits/s，在忽略时隙之间的保护间隔的情况下，每个终端可传输的最大数据率为 R/M bits/s。

(3) 码分多址(CDMA)

与 FDMA 和 TDMA 中各终端在不同频段或不同时间上占用信道不同，在 CDMA 中，各个站同时占用信道的整个频带。各站发送信号的区别在于它们由不同的码所产生，接收站只有使用正确的码字，才能接收到所想要的发送端的信号。

图 7-32 为 CDMA 的原理框图。假设用户数据率为 R_1 b/s(带宽为 W_1)，在传输之前，将用户数据的每一个比特(+1 或−1)都与 G 比特的一个特定的伪随机码字相乘。所谓伪随机码是指虽然由确定的方法(如带有反馈的移位寄存器)产生、但近似具有白噪声性质(码字在非常长的周期后才会重复)的码，也称为**片码**(Chip Code)。在这里，伪随机码也由+1 和−1 构成，但其比特宽度比信号比特窄 G 倍，其带宽 $W \gg W_1$。与信号比特相乘后的伪随机码仍是原来的或极性相反的 G 比特，因此通过相乘信号带宽由 W_1 扩展到了 W。在接收端，如图所示，解调后的基带信号如果和一个与发端相同的伪随机码字相关(G 个比特分别相乘再求和)，会得到一个幅值为 G 的正或负(取决于信号比特的正负)的峰值。如果与解调信号相关的伪随机码字与发送端的不同，则由于码的伪随机性，相关后的输出为接近于零的噪声。

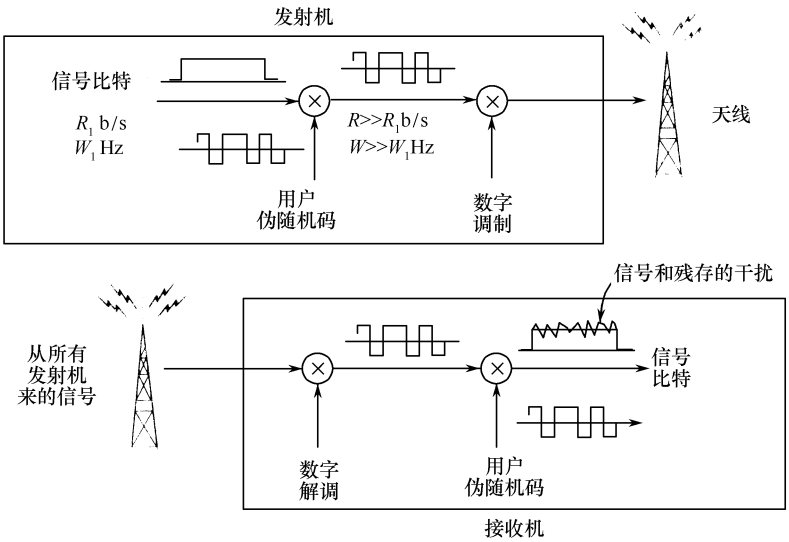


图 7-32　CDMA 原理

由上讨论可以看出，只要每一对收、发终端使用各自独特的伪随机码字，就可以若干对终端同时占用整个信道进行通信，这就是码分多址名称的由来。在 CDMA 中，当信道支持的总数据率为 R b/s 时，每个终端可传输的最大数据率也为 R b/s。

3. 蜂窝移动通信系统的发展

第一代移动通信系统采用模拟技术和 FDMA。在北美的标准称为 **AMPS**(Advanced Mobile Phone System),在欧洲和亚洲的标准称为 **TACS**(Total Access Communication System)和 **NMT**(Nordic Mobile Telephony)。以 AMPS 为例,系统工作在 800～900 MHz,采用频率复用的蜂窝结构,并使用模拟调制技术将电话调制到 30 kHz 的信道上。在第一代移动网上,数据传输需要使用调制解调器,其典型的数据率为 9.6 kb/s。

第二代移动通信网使用数字技术,而在接入方式上则分为 TDMA 和 CDMA 两大类。采用 TDMA 的欧洲标准为 **GSM**(Globe System for Mobile Communications),北美标准为 **IS**(Interin Standard)系列(IS－54/136);由美国 Qualcomm 公司首先提出的 CDMA 成为北美的 IS－95 标准。我国的第二代系统采用了 GSM 和 CDMA 两种标准。

GSM 是一个 TDMA/FDMA 混合系统,它的上、下行信道分别占用 890～915 MHz 和 935～960 MHz 频段(它也可以在 1800 MHz 频段上应用)。它采用时分复用的方式,每个用户的通话周期性地在指定的时隙中传送,最高可达到的数据率为 13 kb/s。**IS－95** 是一个 CDMA 标准,它使用与 AMPS 相同的频段,可以支持 CDMA 与 AMPS 双模工作方式。它所支持的单个用户的最高数据率为 14.4 kb/s。

GSM 是一种电路交换的网络。通用分组无线服务 **GPRS**(General Packet Radio Service)将 GSM 扩展使之支持分组交换的数据业务。在 GPRS 中,8 个 TDMA 时隙合并在一起,组成一个统计复用的信道,为多个用户所共享。当单个用户独占 8 个时隙时,理论上的最高传输速率为 171.2 kb/s。

短信的成功和因特网无线接入的需求,推动了蜂窝网的无线接入从电路交换进一步向分组交换的方向转化;在另一方面,多媒体业务的潜在需求对蜂窝网的传输速率提出了更高的要求。同时,新的频段也被划分给移动通信使用。因此,国际电联于 1998 年提出了 **IMT－2000**(International Mobile Telecommunication－2000)的需求建议书。此后,该项目被称为 **3G** 或 **UMTS**(Universal Mobile Telecommunications System)。一些地区性组织和工业联盟据此分别提出了相关的标准建议。国际电联最终接受的第三代移动通信标准主要有 3 种:WCDMA、CDMA 2000 和 TD－SCDMA。它们都是基于宽带 CDMA 的技术。与 2G 的 CDMA 相比,3G 使用的宽带 CDMA 技术在更宽的频带上扩展信号,具有更大的容量和更高的抗衰落和抗干扰能力。表 7-3 给出了从 2G 到 3G 逐步演进的系列标准。

表 7-3　从 2G 到 3G 移动通信网的演进

	GSM－MAP 核心网	峰值速率	载波带宽
2G	GSM(TDMA)	9.6 或 14.4 kb/s	200 kHz
2.5G	GPRS(TDMA)	170 kb/s	200 kHz
3G	EDGE(TDMA)	384 kb/s	200 kHz
	WCDMA	2 Mb/s(室内)	5 MHz
3.5G	HSDPA/HSUPA	14.4/5.8 Mb/s	5 MHz
	ANSI－41 核心网	峰值速率	载波带宽
2G	CDMA One(IS－95A)	9.6 或 14.4 kb/s	1.25 MHz
2.5G	CDMA One(IS－95B)	115 kb/s	1.25 MHz
3G	CDMA 2000 1X	307 kb/s	1.25 MHz
	CDMA 2000 1xEV－DO	2.4 Mb/s(室内)	1.25 MHz
	CDMA 2000 1xEV－DV	4.8 Mb/s(室内)	1.25 MHz
	CDMA 2000 3X	＞2 Mb/s	5 MHz

表中给出两个系列的标准(表中第二列)演进过程。一个系列是**3GPP**(Third Generation Partnership Project)组织致力的 GSM 的演进,他们定义的 WCDMA 空中接口主要在 GSM－MAP 核心网上应用。这里 MAP(Mobile Application Part)是 GSM 网所使用的识别认证用户和安排话路路由的标准。另一个系列是北美的 **TIA**(Telecommunication Industry Association)致力的 IS－95 的演进,他们定义的 CDMA 2000 空中接口用于 ANSI－41(或 IS－41)核心网,这里 ANSI－41 是与 MAP 作用相同的、在模拟网和 IS 系列网上使用的标准。而 **3GPP2** 组织则与 3GPP 一道定义了 DS(Direct Spread)、MC(Multi-Carrier)和 TDD(Time Division Duplex)等多种模式的空中接口,所有这些模式的空中接口都可以用于上述两种核心网络。

除此之外,**TD－SCDMA**(Time Division-Synchronous CDMA)是我国提出的不经过 2.5 G 的 3G 标准,WCDMA 和 CDMA 2000 都采用频分双工 FDD(Frequency Division Duplex)模式,而 TD-SCDMA 则采用的是时分双工 TDD(Time Division Duplex)模式。所谓 FDD 是指使用分离的两个对称频带分别进行上、下行传输。对于对称业务,如电话,上下行频谱能得到充分利用;而对于非对称业务,如因特网接入,上行数据少而下行数据多,则频带利用率低。TDD 是指采用同一频带进行上、下行传输,上下行各自占有不同的时隙。由于时隙分配的灵活性,特别适合于非对称的分组交换数据业务,下行数据量大可以占用比上行更多的时隙。TDD 的频谱利用率高,但需要在基站同步和较高的峰值/平均功率比下工作。

在 3G 移动通信网中,典型的数据率对于室内静止应用可达 2 Mb/s,对于室外低速和高速应用,则分别为 384 kb/s 和 128 kb/s。因此,真正意义上的多媒体应用只有在 3G 网络上才能得以开展。

国际电联在 2005 年为第四代移动通信网提出了 **IMT advanced** 的需求建议书。3G PP 和 3GPP2 分别提出了 **LTE**(Long Term Evolution)和 UMB(Ultra Mobile Broadband)的发展计划。后来 UMB 被提出者们放弃,LTE 成为各种移动网络演进的统一途径。由于已经推出的 LTE 系统没有达到 IMT-advanced 提出的 1 Gb/s 的峰值速率要求,因此通常被称为 **3.9G**。**LTE-advanced** 是 LTE 继续演进的版本,并已被国际电联批准为 **4G** 的两个系列标准之一(另一系列为 7.9.4 节将介绍的 WiMAX)。LTE-advanced 系列包括 FDD-LTE(由 WCDMA 演进而来)和 TD-LTE(由 TD-SCDMA 演进而来)两种标准。

近年来移动网技术的发展十分迅速,从支持多媒体业务的角度来看,下一代移动网的主要特点为:

(1) 高速移动环境下峰值传输速率为 100 Mb/s,低速移动或静止环境下 1 Gb/s;

(2) 无线资源管理调配方式灵活,支持用户速率动态变化(10 kb/s 到 100 Mb/s);

(3) 数据业务上升为主导地位,完全利用 IP 进行业务传输;

(4) 支持业务分类的 QoS 机制;

(5) 更高的频率利用率和功率效率等。

为了达到上述要求,许多新的技术将被引入。例如,在调制方面,采用 **OFDM**(Orthogonal Frequency Division Multiplexing)技术;在多址接入方面,CDMA 退隐,而 **OFDMA**(OFDM access)和空分多址 **SDMA**(Spatial Division Multiple Access)受到广泛的关注。此外,多输入多输出(**MIMO**)的多天线技术能够利用空间复用增加数据吞吐量,利用其空间分散性扩大覆盖范围,也是极具生命力的技术。

下一代移动网络可以提供 IP 电话、游戏和流媒体服务、视频聊天、手机电视,甚至 HDTV;也可以为智能手机、带有无线调制/解调器(Modem)的笔记本电脑和其他移动设备提供移动超宽带因特网接入,并允许与 WLAN 之间的漫游。

7.9.4 无线城域网(WiMAX)

本节讨论基于 IEEE 802.16 系列的网络。WiMAX 论坛制定的 WiMAX(Worldwide Interoperability for Microwave Access)标准与 802.16 兼容。最初 WiMAX 的目标是作为 xDSL 和同轴电缆的替代方案,提供"最后一公里"的宽带无线接入。后来其服务半径延伸到 50 公里左右,可以覆盖一个城市的区域,成为无线城域网(WMAN)的解决方案。在增强了它的移动性和带宽之后,目前已被国际电联接受为 4G 移动网的标准之一。

1. 系统结构

图 7-33 是 802.16 系统的简要结构图。初期的 802.16 工作在 10～66 GHz 频段,其目标是在基站天线的覆盖范围内,对众多的用户站点提供宽带接入。它是一个一点到多点的协议,用户站点需要通过基站进行连接,相互之间不能直接进行通信。由于工作波长很短,因此必须保证电波在空间能够直线(Line Of Sight,**LOS**)传播,粗略地说,也就是在基站和用户站点天线之间的连线上(实际是连线周围一定区域内)应该不受地球曲率或其他物体的阻挡。显然,在满足这个条件的设计中,基站和用户站点都是固定不可移动的。初期的 802.16 标准在频道带宽为 28 MHz 时,支持 32～134 Mb/s 的数据传输速率。

随后制定的 802.16a 工作在较低的频段(2～11 GHz)上,支持最高 75 Mb/s 的数据率。在此标准下,站点间可以互连,也不限于电波的直线传播。

在图 7-33 中,建筑物内部或园区内的用户可以通过 802.11 WLAN 或 802.16a 连接。802.16 的基站还可以与蜂窝移动通信的基站相连,作为蜂窝系统基站间的传输手段。由此可见,802.16 提供了从替代数字用户线路(DSL)和同轴电缆的"最后一公里"接入(见 7.10 节)到替代有线骨干网的一套完整的无线城域网解决方案。

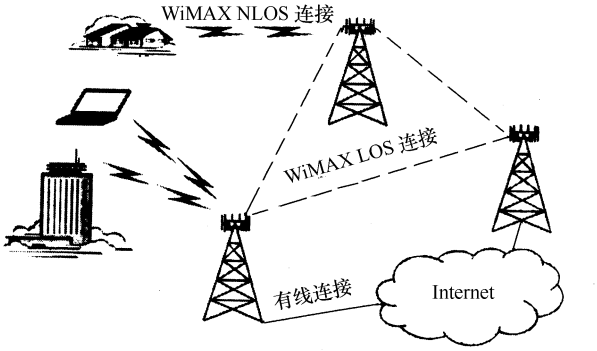


图 7-33 802.16 系统

2. MAC 层和物理层

与 802.11 类似,802.16 只包含对物理层和 MAC 层的定义,其中 MAC 层又分为业务特定会聚子层 CS(Service Specific Convergence Sublayer)、MAC 通用子层 CPS(Common Part Sublayer)和安全子层 SS(Security Sublayer)。CS 层将收到的网络层数据转换成 MAC 数据包,完成网络层到 MAC 层业务流识别符和连接识别符的映射等。802.16 标准支持多种类型数据的传输,为此它定义了两种 CS:ATM CS 和分组 CS,其中分组 CS 可以完成 IPv4、IPv6、以太网和虚拟 LAN 到 802.16 的转换。MAC CPS 完成接纳控制、自动重传、带宽分配、连接建立和管理等功能。SS 提供认证、密钥交换和加密等功能。

802.16 的物理层具有很强的鲁棒性,它提供几种不同形式,以便在不同的环境下能以最少的相互干扰在最长的距离上提供最大的带宽。同时,它还可以针对每个用户站点进行以帧为基础

的调制方式和信道纠错编码方案的自适应调整。例如在小区中心信号较强的区域内，使用支持高传输速率但易受干扰的 64QAM 调制；在小区的中间地带使用 16QAM；在小区边沿地区使用传输速率较低但抗干扰能力较强的 QPSK。基站则在上行和下行信道周期性地发送特定的消息帧向用户说明当前上、下行信道所使用的调制方式和前向纠错码的类型。

802.16 标准使用自适应天线系统，并支持**空时编码**(Space-time Codes)多天线和多输入多输出(MIMO)天线系统。

3. 介质接入和 QoS 支持

802.16 MAC 层是面向连接的。所有的业务，包括无连接业务，都要映射到连接上。每一个业务流和它所对应的连接分别有特定的标识符标识，这为定义基于流或连接的 QoS 参数和带宽分配提供了条件。

802.16 MAC 层在很多方面借鉴了电缆数据传输标准 DOCSIS(见 7.10.3 节)。例如在介质接入方面，上、下行信道的分配都由基站控制。下行(从基站到用户站点)信道采用广播方式，各用户站点只选择接收基站分配给自己的子信道(时隙)上的信号。在上行信道上，用户站点在每个传输周期中的规定时间区间内，向基站提出使用上行信道的请求；基站根据请求分配给接纳的站点以所需的带宽(时隙)。因此，与 DOCSIS 一样，用户上行数据的传输不存在竞争，而只有在规定的请求时间区间内才可能存在多个用户请求消息的冲突。当发生请求消息的碰撞时，802.16 也采用截断二进指数回退窗算法解决。

用户站点需要为每一个连接发出申请上行信道的请求消息，请求中所申请的带宽可以是需要的总带宽，或是在已有带宽的基础上希望再增加的带宽。请求消息可以在规定的有竞争的时间区间内，也可以在已分配的无竞争的时隙中传送。申请附加带宽的消息还可以以捎带(piggyback)的方式传送。基站收到请求后，根据该连接所申请的带宽、该连接上的业务流所需的 QoS 参数(延时、带宽)和当前可利用的网络资源为该连接分配时隙。在有些情况下，基站只为用户站点分配带宽，用户站自己再负责将此带宽分配给本站所有的连接。同时，802.16 还借鉴了 DOCSIS 的业务流调度技术，它能够像 DOCSIS 一样支持：①主动提供传输机会；②实时邀请；③非实时邀请；④尽力而为等不同 QoS 等级的服务(见 7.10.3 节)。按需动态分配带宽和对多类别 QoS 的支持是 802.16 区别于其他无线网络的一个重要特点。

4. 802.16 标准

表 7-4 给出已有的 802.16 系列标准。

表 7-4　802.16 标准

	802.16d(802.16－2004)		802.16e	802.16m
	802.16	802.16d		
频　带	10～66 GHz	2～66 GHz	2～6 GHz	2.6～5.8 GHz
传播方式	LOS	NLOS	NLOS 和移动	NLOS 和移动
传输速率	32～134 Mb/s (28 MHz 信道)	75 Mb/s (20 MHz 信道)	15 Mb/s (5 MHz 信道)	下行：128MB/s(高速)， 1Gb/s(固定) (20MHz 信道)
移 动 性	固定	固定、游牧	固定、游牧、中低车速	固定和低、中、高速
小区半径	2～5 km	7～10 km，最大 50 km (点到点，且取 决于天线高度、功率等)	2～5 km	几十公里

从 802.16e 开始的标准已将终端的移动性考虑在内，因此称为移动 WiMAX。特别是 802.16m 已被国际电联接受为 4G 移动网的标准之一，它支持与无线局域网(802.11)和蜂窝移动网之间的漫游。

5. 无线网络的发展与融合

从整个 7.9 节我们看到，蜂窝移动通信网对用户移动性有很好的支持，它的演进主要体现在数据传输速率的逐步提高。而另外两种无线传输网络，即无线局域网(802.11)和无线城域网(802.16)，也在不断地发展。这两种网络原本具有较高的传输速率，但只支持用户在静止或低速运动状态下工作。由于设计目标越来越趋同和越来越多的相同技术(如 OFDM、OFDMA 和 MIMO 等)的采用，这两种网络对用户移动性支持的程度逐渐趋向于与蜂窝移动网络相匹敌。可以预期，未来这 3 种无线网络的性能将处于类似的范畴，因此可能构成竞争和互补的格局，并最终形成一个全 IP 化的宽带移动网络。

7.9.5 无线网络中多媒体传输的特殊问题

以无线网络(蜂窝网、WLAN、WiMAX 等)作为底层传输机制的 IP 网除了具有一般 IP 网的普遍性问题外，还具有自己特殊的问题。从多媒体信息传输的角度看，这主要表现在如下几个方面：

1. 噪声和干扰引起高的误码率

如 7.9.1 节所述，无线信道上传输的比特错误和突发错误比有线信道上可能高出几个数量级。在有线网络上由于传输误码很少，包的丢失主要由网络拥塞引起的节点缓存器溢出造成；而在无线网络上，丢包则主要由信道的传输错误造成。这使得原有的 TCP 协议在无线网上不再适用，因为利用丢包来判断网络拥塞的 TCP 的拥塞控制功能(见 9.2.1 节)将会极大地降低吞吐量。无线网的固有时延和突发错误也可能引起大量的数据重传。因此在无线网络中要对 TCP 进行较大的改动(见 9.2.5 节)。

除此之外，虽然在无线网络的物理层和数据链路层已经采取差错控制措施，如前向纠错码、自动请求重发等，但误码和丢包仍然是上层多媒体系统需要面对的重要问题。解决这一问题的方式通常包括：打包方式(8.2.3 节)、传输层差错控制(9.3 节)和编码层差错控制(9.4 节)等；同时，将上述措施与下层的机制，如路由或其他物理层和链路层参数，一起进行优化也是一个重要的途径。后者称为**跨层优化**技术。本书对跨层优化没有进一步的讨论，有兴趣的读者请参阅有关文献。

2. 衰落引起信道容量的动态变化

从发射端来的直达电磁波和由障碍物反射产生的散射波相叠加，在空间上形成类似于驻波的变化的电磁场，其场强峰值和谷值之间的距离可以小至 1/4 波长，对于典型的蜂窝系统和 WLAN 而言约为几个厘米。在这样的场中，当接收终端的位置发生移动、或者障碍物移动改变了场强的分布，接收信号的质量会发生相当大的变化。这种衰落效应可以看成为无线信道的容量随时间而随机地变化。在现代移动通信系统的物理层采用好的纠错码的情况下，信道容量在传输一个数据包的时间内可以认为是一个不变的常数；而从传输一个数据包到另一个数据包，信道容量的变化则取决于若干因素，例如接收终端的移动速度。同时对于许多系统而言，如果在某个时刻信道容量显著高于或低于典型值，则至少在数个数据包的传输期间都会保持该值。显然，当移动系统的数据传输速率大于信道容量时产生丢包；而发生一连串的丢包则相当于信号中断。

解决衰落引起信道容量动态变化问题的一种有效方法是，让数据通过相互独立的多种途径到达接收端，如果一种途径的失败概率为 p，两种途径失败的概率则降低到 p^2；当 p 较小时，性能的改善是显著的。因此，几乎所有现代移动通信系统都从不同角度引入了数据传输方式多样性的思想。

(1) 时间多样性。对信号进行分组纠错编码，即将 k 个信息符号加上 $(n-k)$ 个冗余符号构成具有一定纠错能力的符号组进行传输(9.3 节)，可以看作是实现时间多样性的一种方法。此时，分组码分组长度的选择应该使得传输一个分组的时间能够覆盖衰落时间变化的范围。

(2) 频率多样性。衰落是与频率相关的,因此采用多个频带或者一个带宽足以覆盖各种衰落条件的频带进行传输可以提高数据接收的可靠性。例如,802.11 WLAN 所使用的频带宽度就可以满足频率多样性的要求。

(3) 天线多样性。发送终端或接收终端或者二者使用多个天线,这就为信息传输提供了多条空间传输的路径。近年来,多天线技术已经在移动通信系统的设计中受到广泛的重视。

(4) 路径多样性。它与多天线系统一样为数据传输提供多条空间传输路径,不同之处在于它使用位于不同地点的发射机进行发送,这些发射机通过有线网络获得数据。显然,要使这些发射机同步地发射同一数据是困难的,也是浪费的,因此我们更感兴趣的是发射机分别发送互补信息的情况。研究表明[10],将适当的信源编码和纠错编码与多路径传输(包括多天线的情况)相结合可以提高视频传输的质量和鲁棒性,10.4 节将要介绍的**多描述编码**就是一种可以用于多路径传输的信源编码方式。

在应用层上,根据当前信道带宽动态地调整多媒体系统传输码流的码率,也是解决无线信道容量动态变化的一个有效途径。8.6.7 节和 8.8 节将要介绍 3GPP 分组交换流服务和动态自适应 HTTP 流传输就是这方面的例子。

3. 漫游和小区切换引起的包丢失

当移动终端在通信过程中漫游并从一个小区切换(Handoff)到另一个小区时,可能发生短时间的传输路径中断,使得一连串的数据包丢失。在另一方面,当移动终端进入新的小区时,可能由于前后两个小区的接入技术不同(如 WCDMA 和 WLAN)或用户数量的不同,终端的可利用带宽发生变化,或者还可能造成新小区内的网络拥塞,从而导致数据包的丢失。因此在切换过程中减少包丢失,并使视、音频流自适应地平滑调整到新的可利用带宽是研究者目前仍在工作的课题。已经提出的思路包括:①新的网络层移动性管理技术以减少传输路径中断引起的包丢失;②新的传输层技术以避免拥塞并保证数据的可靠传输;③接收终端在小区切换之前对切换时间进行估计,然后通过加大发送端发送速率、或降低接收端播放速率在接收端缓存器中为切换期间准备足够的数据,以避免视、音频流播放的中断。例如,在欧洲手持设备电视标准 DVB-H 中,采用了类似于时分复用的**时间片**技术,每一个时间片以全部带宽突发传送一种业务(例如一个频道)的数据,接收某一种业务的终端只在特定时间片接收和缓存数据,在其他时间片可以对邻近蜂窝进行监测并进行平稳无缝的切换。

4. 终端处理能力和功率的限制

移动通信系统的终端是可移动的,并可以在任何地方和任何时间使用,因此经常以电池作为电源,其功率和寿命受到限制。同时,终端设备体积较小,这就使得它在内存容量和数据处理能力方面受到限制。多媒体通信系统的设计必须考虑到这些限制。

终端功率的限制也使得**功率控制**成为除拥塞控制(9.2 节)和差错控制(9.3 与 9.4 节)之外,在无线**多媒体**系统的 QoS 控制和跨层优化等方面需要考虑的另一个重要问题。

7.10 宽带用户接入网

用户接入网一般指市话端局(本地通信节点)到用户之间的网络。用于多媒体传输的宽带接入网络主要包括本节的数字用户线路、光缆接入、光纤同轴电缆混合接入,以及前面介绍的以太网、无线局域网、3G 及以后的移动通信网和 WiMAX 等。

7.10.1 数字用户线路

在电话网中普遍使用一对铜线构成的本地环路将用户与电话局相连。在铜线构成的早期用户

环路中，线路拼接、线径改变和桥接接头等造成的传输特性的不均匀，使得线路只能用于模拟的低频窄带语音传输。到了 20 世纪 80 年代，由于自适应数字滤波和大规模集成电路技术的发展，使自适应回波抑制和均衡技术进入了实用化阶段，能够对线路传输的不均匀性进行精确的补偿，从而形成了利用铜线传输高速数据的**数字用户线路** DSL(Digital Subscriber Line)。在 DSL 上，除了 3.4kHz 以下的语音通信外，对高端频带(10kHz 至几十 MHz)的利用形成了一系列的宽带接入技术，统称为 xDSL。

1. 高速数字用户线路(HDSL)

HDSL(High-SPeed DSL)通常用来提供双向的 E1(2 Mb/s)接入，服务范围大约为 3.6 km。一个 HDSL 用户使用 2 对双工速率为 1176 kb/s 的线路合起来构成 E1。使用 2 对线路的原因是为了降低每一对线路上的工作频率，因为工作频率越高，通过线间耦合电容形成的串话和信号能量在传播过程中的衰耗越严重。

近年出现的 **SDSL**(Symmetric DSL)可以在单对线上提供对称的 E1 接入，传输距离 5.5 km。

2. 非对称数字用户线路(ADSL)

ADSL(Asymmetrical DSL)是在一对线上进行非对称双向传输的技术。由于只需要一对电话线，因而比较适于家庭使用。如图 7-34 所示，利用频分复用的方法，常规的电话仍使用原来的低频段，而高速数据则调制到高频段上传输，其中下行方向的数据率高，而上行方向的数据率较低。对于一般的用户而言，信息查询是所需要的主要业务，因此，这种非对称的信道比 HDSL 更为经济。

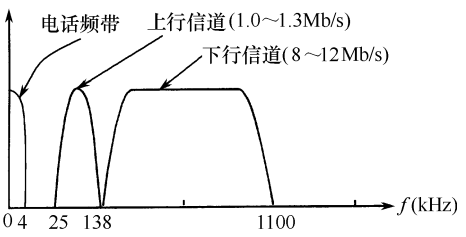


图 7-34　ADSL 频带分配

图 7-35(a)给出了一个 ADSL 回路的简要方框图。在中心(电话局)的 ADSL 机架中完成下行、上行数据的调制/解调，以及与电话的混合与分离；在用户端的网络终接单元中利用滤波器将电话和数据信号分离，并完成上行、下行数据的调制/解调。数据信道与终端设备(电视机或微机)的接口，对于电视机而言，是机顶盒 STB(Set-Top-Box)；对微机，则可以直接接入。

ADSL 上行占用频带 26～137.8 kHz，下行占用 138～1100 kHz，其上行信号速率为 1.0～1.3 Mb/s，下行速率可达 8～12 Mb/s；传输距离为 4 ～8km。ADSL 和所有的 xDSL 一样，其实际传输速率会受传输距离和线路质量的影响。

ADSL-Lite 是一种简化的 ADSL 标准，也称为 G.Lite，Universal ADSL。它占用的频带在 25～500 kHz，其上行和下行速率分别为 32～512 kb/s 和 64 kb/s～1.5 Mb/s。如图 7-35(b)所示，它在用户端不使用网络终接单元作为分离器，降低了成本，简化了安装手续，但此时妥善地解决对电话的干扰是需要考虑的主要问题。

ADSL2 与 ADSL 占用的频带相同。它的上、下行传输速率增加不多，分别为 1.3～3.5Mb/s 和 12 Mb/s；其主要改进在传输距离加长，以及有桥接头和受射频干扰等情况下传输性能的改善等方面。

ADSL2＋和之前的 ADSL 各种标准不同，它将上频带由 1.1 MHz 扩展到 2.2MHz，因此其传输速率增加了一倍，上行达到 1.3～3.3 Mb/s，下行速率最高可达 24 Mb/s。

3. 甚高速数字用户线路(VDSL)

VDSL(Very-High-Speed DSL)常常用在光缆到路边、或光缆到企业(见 7.10.2 节)的网络中，作为最后一段连接到用户终端的铜线。它占用的频带较高(25 kHz～12 MHz)，传输距离为 500 m～1 km。传输速率在双向对称应用时可达 25 Mb/s；在非对称应用时，下行可达 52 Mb/s，上

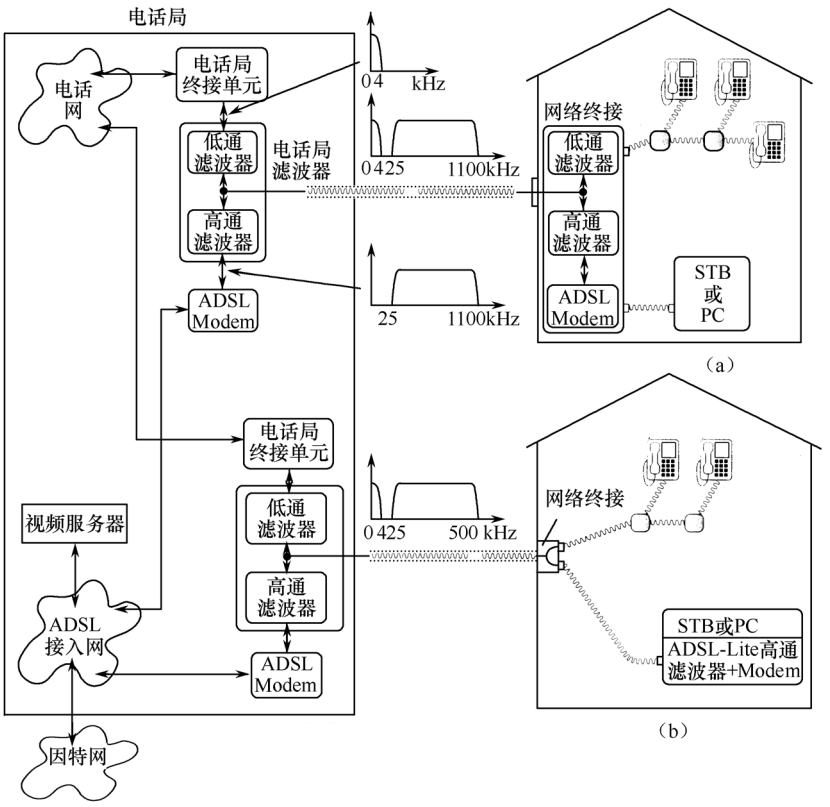


图 7-35 (a)ADSL 结构;(b)ADSL－Lite 结构

行为 16 Mb/s。

UDSL2 将占用频带扩展到了 30 MHz,在 0.5 km 的传输距离上,其上、下行速率均可达到 100 Mb/s。当距离增加到 1 km 时,速率为 50 Mb/s;增加到 4～5 km 时,下行速率仍可达 1～4 Mb/s。因此,VDSL2 不像 VDSL 那样,只能限制在短距离的传输上。

7.10.2 光缆接入

光缆直接铺设到用户家中、或接近用户的路边,可以为用户提供高速率(25～51 Mb/s)和高信噪比的信道,这是很吸引人、却也是造价最高的一种方案。由于光缆铺设的拓扑结构直接影响到造价,因此本节重点讨论这一问题。

1. 单星形结构

单星形结构如图 7-36 所示,其中 OLT(Optical Line Termination)是设在交换局内的光线路终端,ONU(Optical Network Units)为用户端的光网络单元,V_5 是交换机与接入网之间的接口标准。由图可见,光缆从 OLT 辐射出来,构成到用户的点到点的线路。

由于在 ONU 端光电转换设备的费用较高,因此单星形结构仅适合于向大企业单位提供服务。此时每个 ONU 通过一个用户端的复用器与众多的电话线相连。如果向个人用户提供服务,由于每条光缆所连的电话线数少,平均每根光缆所需的光电转换设备的费用则升高;在极端的状况下,一条电话线需要一条光缆相连,这显然是不经济的。

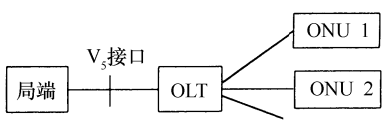


图 7-36 单星形结构

2. 有源多星形结构

有源多星形结构如图 7-37 所示。在这种结构中，多个 ONU 可以共用一根交换局端的光缆。在有源复用节点上，从多个 ONU 来的光信号转换成电信号，然后复用在一起，并转换成光信号以共用一根局端的光缆。

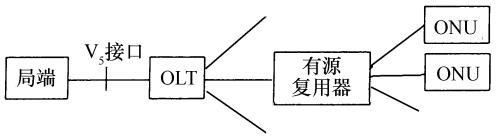


图 7-37　有源多星形结构

在有源多星形结构中，光缆的费用显然会降低，但却增加了有源节点的设备、安装、维护及供电等方面的投资。同时，由于在 OLT 和 ONU 之间插入了电传输部分(有源复用器)，形成限制光缆容量扩充的瓶颈。当需要在光缆上提供更高速率的服务时，必须改造复用节点。

3. 无源多星形结构

无源多星形结构的出现使得光缆接入真正成为一种现实可行的手段。图 7-38 所示为它的基本结构，它和有源多星形结构一样，多个 ONU 可以共用一条局端光缆，而且从 OLT 到 ONU 全部是光通路，一个光发射机和一个光接收机就可以支持所有的用户终端，这进一步降低了铺设网络的成本。

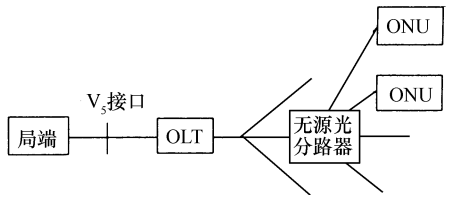


图 7-38　无源多星形结构

在无源星形结构中，下行方向通过无源分路器将光信号分配到各个 ONU；在上行方向上，分支器又将从各个 ONU 来的信号汇集到共同的 OLT，形成一个树形的点到多点的双工传输系统。

在无源星形结构上实现一点到多点的通信需要采用复用技术。OLT 利用 TDM(或波分复用 WDM)技术将要送到各 DNU 的信号复用成一个信号，然后采用广播方式传送下去，各 ONU 则选择自己对应的频道接收。将各 ONU 的上行信号复用到一起要困难一些，因为这要求各 ONU 必须是同步的，否则，不同 ONU 所发出的信号可能重叠在一起。实际上在时分多址接入(TDMA)的移动通信中也有同样的问题，处于不同位置上的手机所发出的信号也是通过时分复用返回到一个公共的基站的。移动通信中利用一个称之为测距协议(Ranging Protocol)的方法来避免各手机返回信号的混叠，类似的方法也可以借用到 ONU 上行信号的复用中去。通常，使用这种方法使距离 OLT 20 km 范围内的 ONU 达到同步是不困难的。由于光缆中信号的衰耗很小，因此，无源多星形结构接入的服务范围可达 20 km。

上行和下行各使用一根光缆是最简单的实现双向通信的方法，这称为**空分双工**。而在一根光缆中传输两个方向的信号就必须有一定的方法将两个方向的信号区分开来。最常见的是利用不同波长，例如 1 310 nm 和 1 577 nm 的光波传送不同方向的信号，这称为**波分双工**。

4. ONU 的配置

如果用户端的 ONU 只对一个家庭提供服务，则称为光纤到家庭 FTTH(Fiber-to-the-Home)；如果 ONU 向若干条线路提供服务，则构成光纤到企业 FTTB(Fiber-to-the-Business)、光纤到路边 FTTC(Fiber to-the-Curb)等。显然后面几种方式比 FTTH 更为经济。当几个用户线共用一个 ONU 时，需要在 ONU 中进行复用/解复用。从 ONU 到最终用户点之间通常用不长的铜线连接。

7.10.3　光缆－同轴电缆混合接入

1. 网络结构

早期的有线电视 CATV(Cable Television)网络是由同轴电缆构成的多枝杈的大型树状网络，

有线电视中心(行业术语称之为**头端**,即 Headend)在树根处,枝杈伸向用户家庭。由于电信号在同轴电缆中衰耗较大,特别是当利用的频带较宽时,高频的衰耗更为明显,因此每隔一定的距离需要加入放大器提高信号的增益,以增大传输距离。这些数目众多的级联放大器往往使网络工作的可靠性降低。

由于光缆中信号损失很少,在 CATV 网的干线上使用光缆,如图 7-39 所示,可以去掉影响信号质量和可靠性的干线放大器并消除电源供给问题,从而在很大程度上提高了网络传输的稳定性。这种结构称为**光缆－同轴电缆混合方式** HFC(Hybird Fiber/Coax)。光缆从头端呈星形、或环形辐射出去,在住宅小区附近通过混合节点(Hybird Node)与通往各个家庭的树状小型同轴电缆网相连。一个混合节点所服务的小区称为一个服务区。头端之间、头端和主头端之间还可以通过光缆环(如 SDH 环)相连,以构成更大规模的网络。与早期的 CATV 网相比,HFC 网不仅可靠性增加,而且由于光缆的容量很大,而每个服务区内用户数较少,如 500 户左右,因此有利于向宽带接入网的方向发展。早期的大型树状网上往往有 1～2 万的用户,如此多的用户共享有限的同轴电缆带宽,显然是不适合作为多媒体的宽带接入途径的。

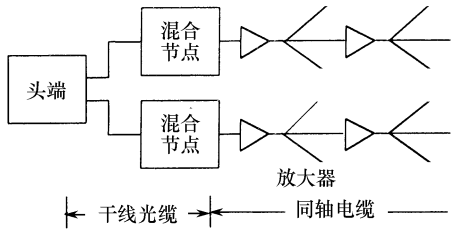


图 7-39　光纤－同轴电缆网络

2. 频带分配

在 HFC 网上除了原有的模拟有线电视服务之外,可以采用频分复用的方法加入其他的服务(如数据、电话、VOD 等)。HFC 可利用的频带主要由同轴电缆的特性决定。现代同轴电缆的带宽可达 750 MHz、860 MHz 甚至于 1 GHz,传输距离为 300 到 600m。图 7-40 给出了一种典型频带分配方案。在50～550 MHz频段传送 60 个频道的模拟电视,550～750 MHz 频段传送下行(从头端至用户)的数据,5～42 MHz 频段传上行的数据,750～1 000 MHz 频段预留给未来的双向业务。值得注意的是,HFC 的上行和下行频段是混合节点之后(服务区内)的所有用户所共享的,这一点与 ADSL 中每个用户独占一对双绞线的情况不同。

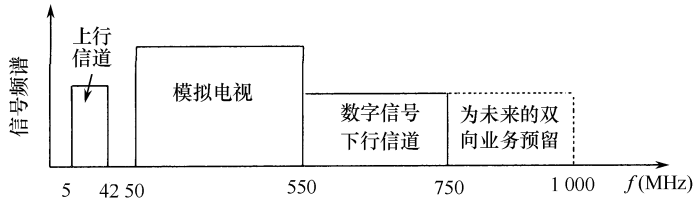


图 7-40　典型的 HFC 频带分配

3. 双向数字信道

有线电视网是单方向的模拟广播信道,在提供交互式多媒体服务时需要将它改造为双向数字信道。双向改造包括在同轴电缆段将单向放大器改造成双向放大器,以及在光缆段增加上行的光端设备和上行光缆;数字调制技术则用来使上、下行数字信号适配进原来的模拟信道。

上行信道使用同轴电缆的 5～42 MHz 频段,这一频段对无线设备(如无线广播、寻呼和移动台)和家用电器产生的干扰很敏感,而树形的分支结构又使在各用户端受到的干扰在会聚点叠加,这称为"**漏斗效应**"。漏斗效应使上行信道的信噪比降低。采用抗干扰能力较强的先进的调制技术、以及在用户接口处设置滤波器阻断未使用的频段等方法可以部分地解决这一问题。另一种解决方案是划分频带时避开最易受干扰的频段而选用较高的频段,例如 42～108 MHz,作为上行信

道;或者减少服务区内的用户数以减少引入上行信道的干扰。后者同时可以使每一个用户占用的上行信道带宽增加,有利于采用抗干扰能力较强的调制技术。上行信道在许多现有的商业系统中采用 QPSK 或 16QAM 调制。由于上行信道频带较窄,又不像下行信道那样有规范的频道划分,为了有效地利用频带资源,常常将该频段划分成一系列子频段,使网络运营者可以针对不同业务进行分配,也可以动态地进行频率分配。典型的子频段宽度和比特率如下:

160 kHz:320 kb/s	1.28 MHz:2.56 Mb/s
320 kHz:640 kb/s	2.56 MHz:5.12 Mb/s
640 kHz:1.28 Mb/s	5.12 MHz:10.24 Mb/s

HFC 的下行信道一般采用多进的残余边带调幅(VSB/AM)、多进正交调幅(QAM)或正交频分复用(OFDM)技术将下行数字信号(点播电视、数字视频或音频广播、数据、电话等)调制到中频,再经高频调制后进入模拟电视的频道。其中 QAM 的频谱利用率较高,例如 64 QAM 可以达到 5 b/s·Hz,在一个 8 MHz 的标准模拟频道内可以传输速率约 40 Mb/s 的数据,也就是说,在原来一路模拟电视的频带中,可以传 6 路 6 M MPEG2 的数字电视。

如前所述,HFC 的上行和下行信道都是共享的,下行信道按固定频道划分,其频带资源可以由头端设备进行分配,而由上百个用户共用的上行子信道则存在着多址接入的问题。此外,利用 HFC 网络传送多媒体数据时,其上行和下行信道均需要采用前向纠错编码来提高抗干扰的能力。

4. 系统结构

图 7-41(a)给出了一个支持交互式多媒体应用的 HFC 系统的简要方框图。该系统仍保持原来传输模拟(或数字)电视的功能。模拟(或数字)电视通过地面广播、卫星广播或光缆线路传输到头端,在头端经过频率转换送进有线电视的相应频道。在用户端,模拟电视信号直接进入电视机,数字电视信号则需经过机顶盒 STB(解调和解码)转换成模拟信号再接入电视机。

在提供宽带数据业务,如 IP 电话、因特网接入时,头端有一个**电缆调制解调终端设备 CMTS**(Cable Modem Termination System),其简要方框如图 7-41(b)示,其中 H.323 是 IP 电话的一个标准(见 8.3.3 节)。在 CMTS 中,来自 LAN 或 WAN 的数据经过调制/解调以适应 HFC 上行和下行传输信道的频带要求。在用户端完成对应功能的装置称为**电缆调制解调器 CM**(Cable Modem),它的输出端为一个、或多个以太网接口[见图 7-41(a)]。CMTS 和电缆调制解调器的功能相当于一个路由器,用户终端像通过永久性虚连接直接接到这个路由器的本地以太网段上一样。在下行方向上,IP 包在指定的频道上以广播方式传播,所有的 CM 都可以接收到。但在上行方向上,多个 CM 在争用信道时,距离 CMTS 较远的 CM 不能侦听到比它距 CMTS 更近的 CM 的发送情况,因此,一般的 MAC 协议如 CSMA/CD 和控制令牌,不能在这种情况下应用。**DOCSIS**(Data-Over-Cable Service Interface Specification)/IEEE 802.14 是电缆调制解调系统广泛采用的标准。对于这个标准我们后面将会简要地讨论。

图 7-41(a)的系统在提供交互式视频业务,如交互电视、VOD 等时,头端的视频服务器能够同时输出多个 MPEG2 的码流。这些码流经过复接器、数字调制(如 64 QAM)和高频调制到指定的频道上,然后与其他频道信号混合进行传输。在接收端,机顶盒接收该频道上的高频信号将其解调成数字基带信号,并完成解压缩和 D/A 变换,最后将恢复出的模拟信号送到电视机。用户通过输入设备(如遥控器)发出的交互/点播命令,需要通过上行信道送给视频服务器。如果机顶盒中内嵌了电缆调制解调器,则交互命令可通过上行数据信道传输。

5. 电缆系统数据传输标准

我们在这里简要介绍 **DOSIS/IEEE 802.14** 标准的原理。

在电缆数据传输中,CM 之间不能相互直接通信,需要经过 CMTS 转发。同时,CM 是一个无硬

盘设备，需要 CMTS 对它进行配置和管理，此外 CMTS 还负责对上行多址接入进行控制。

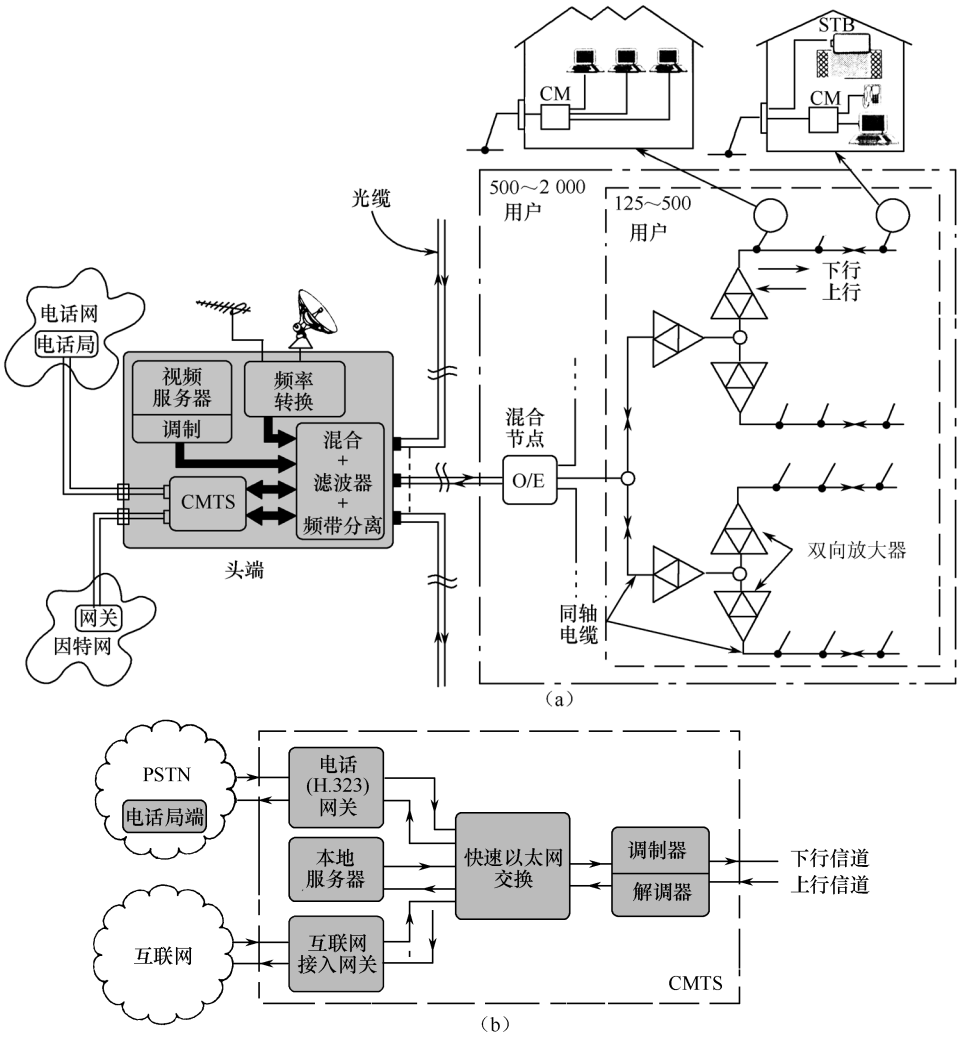


图 7-41　(a)HFC 系统构成；(b)CMTS 框图

为了实现对 CM 的配置，除了连接在 CM 上的用户终端有各自的 MAC 地址外，CM 本身有自己的 48 位 MAC 地址。图 7-42 表示出电缆数据传输的协议栈。图中双箭头的实线表示逻辑路径，双箭头的虚线和点画线表示物理路径。为了使 IP 层以下的层对用户都是透明的，在 CM 中对 IP 包的传递在 MAC 子层进行，因此 CM 的作用相当于桥；而在 CMTS 中，IP 包的传递在 IP 层进行，因此 CMTS 对于所有连接在电缆上的用户终端而言相当于因特网的接入网关。CM 和 CMTS 分别还有 MAC 层和 IP 层以上的协议栈，其作用是完成对 CM 的配置和管理。在 CM 完成注册和配置之后，CMTS 给予它一个服务标识 SID(Service Identifier)。

如前所述，CM 不能完全侦听到所有其他 CM 的发送情况，需要 CMTS 对上行多址接入进行控制。首先，上行信道按照固定的时间间隔分成一系列传输周期，每个周期又分为多个等间隔的小时隙。在一个传输周期内，有一段时间(包括若干小时隙)作为各 CM 申请上行信道的时间，称为**REQ 区间**；其他时间用来传输上行数据。当 CM 需要传输数据时，它在当前周期的 REQ 区间内向 CMTS 发出要求一定带宽的请求。CMTS 根据接纳算法决定是否接纳请求并分配给它一定数量的小时隙，这个信息连同该 CM 的 SID 一起在下行信道中广播。提出请求的 CM 获知分配给它的

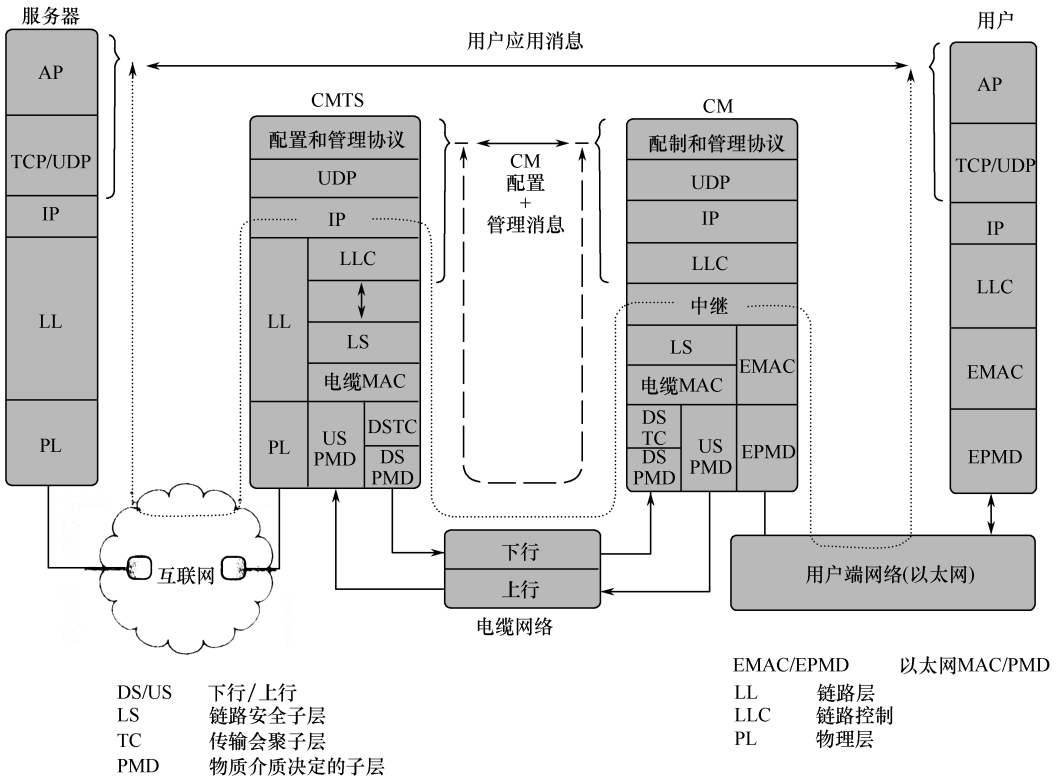


图 7-42　CMTS/CM 协议栈

时隙位置后，在下一周期的相应时间内发送自己的数据。如果获知分配给它的时隙数为零，则表示尚未被接纳，还需要等待再一个周期。

HFC 网络通常覆盖一个相对广阔的地域，信号在电缆中的传播时间不能忽略。由于距 CMTS 的距离不同，每个提出请求的 CM 收到返回信息的时间也不同。要使各 CM 发送的数据在到达 CMTS 时正好准确地处在分配给它的时隙中，这就要求：①各 CM 的时钟与 CMTS 的主时钟同步；②每个 CM 必须事先知道它到 CMTS 的双程延时。同步和测距都需要在 CM 传输数据之前的初始化过程中完成。这里的问题和解决方案与 7.10.2 节中介绍的无源多星形结构上行信道类似。

在 REQ 区间内，可能有多个 CM 发出请求，因此必须有竞争算法来解决冲突。电缆数据传输中使用的竞争算法与以太网类似，也属于 7.6.1 节中介绍的截断二进指数回退窗算法。当 CM 收不到 CMTS 返回给自己分配时隙的消息时，说明 CMTS 没有收到它的请求，因此它等待（回退）一个随机的时间之后，再尝试发送。需要注意，这里与以太网不同的是，发生冲突的是请求消息，而不是数据；数据是在 CMTS 分配（预留）的时隙中传送的。

由于数据传输带宽（时隙）是由 CMTS 统一分配的，因此电缆数据传输系统可以支持不同级别的 QoS。除了“尽力而为”的服务外，它还支持以下服务：

（1）主动提供传输机会（Unsolicited grant）：为实时媒体传输周期性地预留固定的时隙，适于固定包长的实时媒体的传输；

（2）实时邀请（Real-time polling）：CMTS 周期性地向传输实时媒体数据的 CM 发出预留带宽的邀请，适于变包长的实时媒体的传输；

（3）带有静默检测的主动提供传输机会（Unsolicited Grant with Activity Detection）：在为一个 CM 主动提供传输机会时，如果 CMTS 发现该 CM 没有利用这些传输机会，例如正处于语音的

静默期，则停止分配给它传输机会。如果 CM 再度有数据发送，它向 CMTS 发出请求，CMTS 恢复对它的主动提供传输机会的服务；

(4) 非实时邀请(Non-Real-Time Polling)：此服务用于大数据量，例如大文件的传输。CMTS 周期性地向 CM 发出预留带宽的邀请(避免了请求消息的竞争)，这可以在网络负荷较重时，缩短传输大文件的时间。

7.11 下一代网络

在对各种核心网和接入网技术有了基本的了解之后，本节扼要介绍将这些网络综合在一起构成下一代网络的思路。

7.11.1 网络融合模型

许多年来，我们所获得的各种信息服务是通过不同的网络提供的，移动电话、普通电话、计算机业务和电视分别由蜂窝移动网、电话网、计算机网和广播电视网所支持。每个网络有自己独立的接入系统、骨干传输和交换系统，这种提供多种业务的方式如图 7-43(a)所示，称为**垂直模型**。垂直模型显然不利于提供各种媒体集成在一起的多媒体业务。正如 1.5.1 节所总结的，近年来通信网、计算机网和广播电视网络融合的趋势正在改变这种模型，不同类型的传输技术和接入技术将逐渐统一到一个框架下，各种业务都在 IP 之上提供。图 7-43(b)给出了下一代网络的示意图，图中不同的终端通过不同的接入网接入到统一的传输网络上，各种业务都通过这个统一的传输网络传送。图 7-43(c)给出了(b)的层次结构，这种模型称为**水平模型**。水平模型的好处是，应用、控制和传输完全分离，各个层可以独立于其他层进行演变和改进。我们在 7.8.3 节中讨论 GMPLS 时已经看到，控制层和连接层(也称承载层)分离在采用不同技术网络的融合中所具有的重要意义。

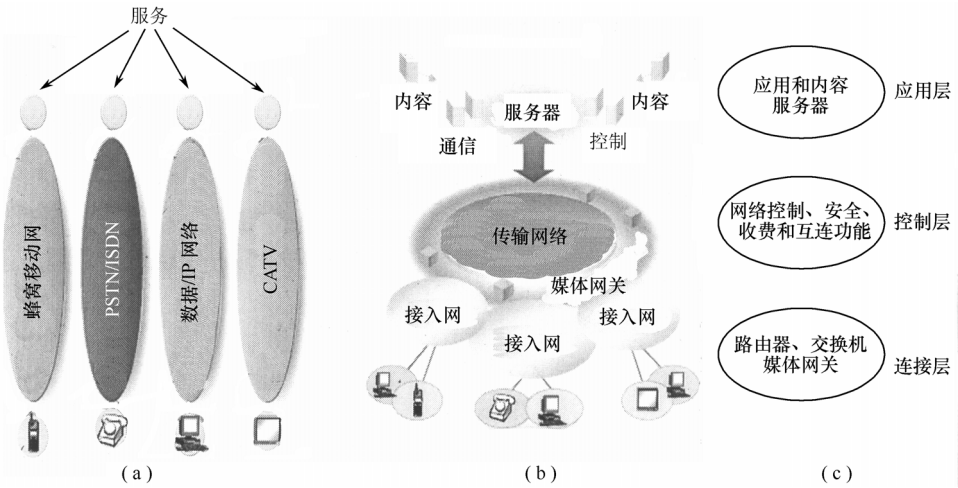


图 7-43 网络融合的垂直模型和水平模型

7.11.2 IP 多媒体子系统(IMS)

IP 多媒体子系统 IMS(IP Multimedia Subsystem)是 3GPP/3GPP2 为 3G 移动网提供移动多媒体业务提出的一个标准。它体现了水平模型的思想，由于采用了统一的控制层，使得采用不同传输技术的承载网和接入网相对于业务层是透明的，从而可以通过一个统一的传输平台向用户提供综合的多媒体业务。由于 IMS 网络结构的灵活性，该标准已被包括 ITU－T 在内的其他国际组织

所接受并扩展到有线网络，使之正在成为网络融合和核心控制层发展的一项重要技术。

1. IMS 网络结构

IMS 核心网络是基于 IP 的，一个 IMS 用户终端需要支持 IPv6（或 v4）和**会话发起协议 SIP**（Session Initialization Protocol）。SIP 是 IETF 制定的信令协议，我们将在 8.4 节中讲述。IMS 终端可以通过 xDSL、有线电视网、以太网、WLAN、WiMAX 或 3G 移动网等多种形式接入。普通电话机、H.323 终端（见 8.3.3 节）和非 IMS 的 IP 电话终端则需要通过相应的网关接入。

图 7-44 为 IMS 网络结构的简化示意图。最高层的应用服务器提供各种增值业务。控制层中的归属用户服务器 **HSS**（Home Subscriber Server）保存着有关用户和应用服务器信息的数据库，完成用户的认证和鉴权。由于 IMS 是支持漫游的，所以 HSS 还负责提供有关用户物理位置的信息。呼叫控制功能 **CSCF**（Call Session Control Function）是控制层最重要的部分，它负责处理 SIP 信令。当收到发送端请求通信的 SIP 消息后，CSCF 从 HSS 获得必要的定位信息，建立起收、发端的连接，并提供会话控制服务；在通话后负责释放连接。

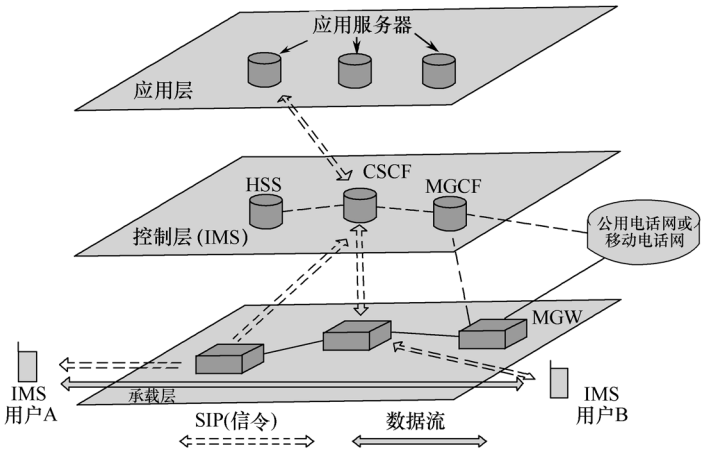


图 7-44　IMS 网络结构

图中用带箭头的双虚线表示建立 A、B 通话的 SIP 信令的传输路径；用带箭头的双实线代表媒体流的传输路径，我们看到媒体流不经过信令路径而是通过承载层直接在 A、B 之间传送。公共电话网、移动电话网或其他电路交换的网络需要通过网关才能与基于 IP 的 IMS 网络相连。图中媒体网关 **MGW**（Media Gate Way）负责媒体流的互通，并在两边网络协议和媒体格式不同时（如音频或视频压缩格式不同）进行必要的转换；**MGCF**（Media Gateway Control Function）则通过 ITU－T H.248 协议完成对媒体网关的控制。

2. IMS 支持的业务

通过 IMS 实现的网络融合使得运营商可以通过统一的业务层和控制层向不同接入网上的用户提供多种业务，也就是说，电话、移动电话、电视以及 1.4 节介绍的各种多媒体业务不再必须由多个运营商通过多个独立的网络提供。不仅如此，网络“无缝”的连接还使得用户有可能在一项通信过程中使用不同的终端设备跨越不同的接入网获得不中断的服务。1.5.2 节中曾举出过这方面的例子。

IMS 的网络融合框架激发了应用和业务的融合。IMS 允许将各种业务（包括未来的新业务）作为一个整体来进行管理，因此可以期望，未来可能出现更为复杂的多媒体应用。

7.11.3 下一代网络框架

2004 年 ITU－T 在它的两个建议中对下一代网络 **NGN**(Next Generation Network)的基本特征作了定义[11,12],其中主要包括:①它是一个包交换的网络;②具有 QoS 机制;③与业务有关的功能独立于底层与传输有关的技术;④能够让用户自由地通过有线和无线宽带网络接入他们想要的业务;⑤支持固定和移动的终端,为用户提供无处不在的一致的服务。在此基础上,许多国际标准化组织和产业联盟进行了对 NGN 的研究,其中比较著名的结果有欧洲电信标准研究院 ETSI(European Telecommunication Standard Institute)TISPAN 小组提出的以 IMS 为核心的架构。ITU—T的 FGNGN(Focus group on NGN)小组吸取了上述成果并于 2005 年发布了有关 NGN 框架研究的第一个版本[13],后续的工作现在由 ITU－T GSI(Global Standard Initiative)小组继续进行。

图 7-45 为 ITU－T 和 TISPAN 提出的 NGN 功能框架的示意图。其中业务层主要包括 IMS 核心子系统、PSTN/ISDN 仿真子系统、流媒体子系统和其他 NGN 业务子系统,以及可以应用于各子系统的公共功能,如应用接入、计费、用户、安全和路由的管理。传输层为用户提供基本的 IP 连接能力,其中的网络附着控制功能 **NACF**(Network Attachment Control Function)和资源与接纳控制功能 **RACF**(Resource Admission Control Function)屏蔽了 IP 层以下的接入网和核心网所使用的具体传输技术,并提供 QoS 能力。

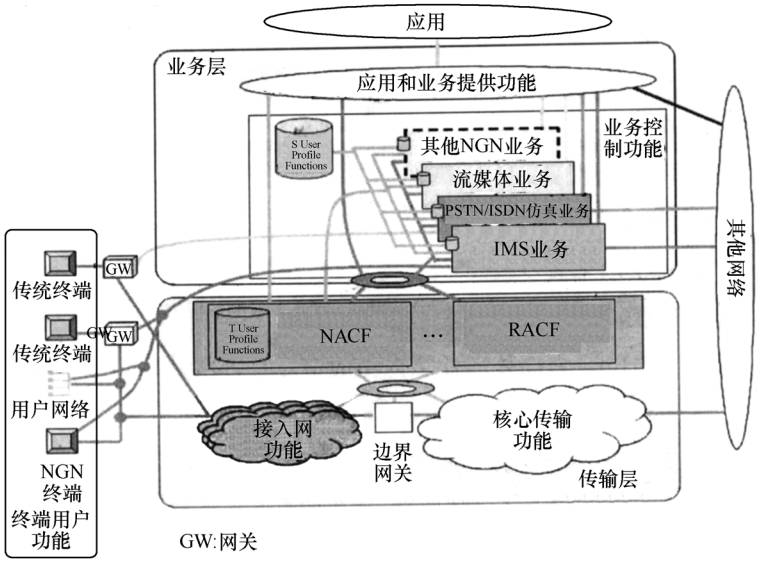


图 7-45 NGN 框架

由图看出,整个框架以 IMS 技术为核心,各种接入网和核心网技术都集成在统一的框架下,所有的业务都通过 IP 提供(其中传统的窄带业务是通过电路仿真系统接入的)。这个框架为我们理解下一代网络给出了一个基本的概念,也帮助我们对本章讨论的各部分内容在下一代网络中所处的位置有一个清楚的认识。

习 题 七

1. 什么是传输延时、传播延时和端到端的延时?多媒体会议业务的实时性要求针对的是以上三种中的哪一种延时?
2. 从通信建立时间、传输延时、延时抖动、带宽利用率、对实时业务的支持、包传输次序及丢失和 QoS 保障等方面对下列网络进行比较:(1)电路交换与分组交换网络;(2)面向连接与无连接网络。

3. 假设欲在北京、天津、上海、广州、重庆、武汉、沈阳和乌鲁木齐 8 个城市之间通过 DDN 或 N－ISDN 线路召开电视会议，并以北京为主会场。而已购买的 MCU 设备每个只有 4 个端口，试做如下设计：
(1)完成上述业务的网络逻辑结构图；(2)设备安装地点(城市)和线路连接图。
4. 你对计算机网络中的桥、网关和路由器的作用有哪些了解？
5. IP 多播协议 IGMP 的工作方式为：①主机通过包括本地网上所有站点的多播地址发送一个 IGMP 消息说明自己要参加哪个组，本地多播路由器收到此消息后，将此信息分发给其他多播路由器以建立多播路由；②本地多播路由器周期性地检查本地网上的主机是否仍在组内，如果在连续几次检查中，没有主机应答表示它仍在组内，则本地路由器停止向其他多播路由器分发本地网组成员的信息。问：
(1)为什么多播路由器需要了解本地网上有无组成员？
(2)该协议如何支持组成员的动态变化？
(3)IGMP 消息采用的是无 QoS 保障的 IP 报文，如果请求消息丢失，会发生什么情况？如果应答消息丢失，又会发生什么情况？
(4)如果一个主机与两个子网相连，则它只需要在其中一个子网上加入多播组。以一个跨子网的电话会议为例，说明为什么。
6. 请对 IP 网的 Intserv 和 diffserv QoS 保障机制的复杂度和性能做出评价。它们能够提供定性的还是定量的、确定的还是统计意义上的 QoS 保障？
7. 你如何理解 RSVP 作为信令协议在 DS 服务中的应用？
8. 假设一台主机与专用 ATM 交换机的一个 155 M 端口相连，在下述两种情况下，主机总线上的数据率需占总线吞吐量的百分之多少才能充分利用 ATM 接口所提供的速率？
(1)主机为使用 PCI 总线的微机；(2)主机为使用 VME 总线的工作站。
9. 业务流的概念是什么？这一概念是如何在 IPv6 中得到应用的？
10. 在 IPOA 中，通过 AAL5 完成 IP 数据报到信元的转换。当一个 AAL5 包(IP 数据报＋AAL5 尾)不能装进整数个信元时，需要在最后一个信元中(在 AAL 尾之前)填充足够的零。假设 ATM 信元丢失概率为 p 时，分别计算 576 字节、1 500 字节、4 500 字节和 9 180 字节长的 IP 报文丢失的概率。在 IP 报文为 1 500 字节时，采用 AAL1 还是 AAL5 传输效率高？
11. 什么是半双工和全双工工作？吉比特以太网和传统的以太网在哪些方面相同，在哪些方面不同？
12. 从图 7-34 和图 7-40 来看，能否认为通过 HFC 用户可以获得比 ADSL 高的带宽？为什么？

参 考 文 献

[1] F. Fluckiger, Understanding Networked Multimedia Applications and Technology, London: Prentice Hall, 1995
[2] [美]F. Halsall 著．多媒体通信．蔡安妮，孙景鳌等译．北京：人民邮电出版社，2004
[3] N. F. Mir, Computer and Communication Networks, Prentice Hall, 2007
[4] F, Le Faucheur, *et al*., Multiprotocol label switch(MPLS) Support of differentiated sevecise, RFC 3270, 2002
[5] D. Awduche, *et al*, Requirements for traffic engineering over MPLS, RFC 2702, 1999
[6] E. Crawtey *et al*., A Framework for QoS-Based Routing in the Internet, RFC 2386, 1998
[7] [美]T. H. Cormen 等著．算法导论(第二版　影印版)．北京：高等教育出版社，2002
[8] B. Jamoussi, *et al*, Constraint-based LSP setup using LDP, RFC 3212, 2002
[9] D. Awduche, *et al*., RSVP-TE: Extensions to RSVP for LSP tunnels, RFC 3209, 2001
[10] S. Mao, et al., "Multipath video transport over ad hoc networks," IEEE wireless communication, Aug. 2005, 42-49
[11] ITU－T Rec. Y. 2001, General overview of NGN functions and characteristics, 2004
[12] ITU－T Rec. Y. 2011, General principles and general reference model for next generation network, 2004
[13] C. Lee and N. Morita, "Next generation network standards in ITU－T," 1st Int. workshop on broadband convergence network, 2006

第 8 章　多媒体通信终端与系统

8.1　传输层协议

8.1.1　应用层分帧和集成层次处理

在计算机通信中往往采用层次化的协议结构，开放系统互连(Open System Interconnection，OSI)的七层参考模型(OSI-RM)就是一个典型的例子。在这个模型中，每一层实现一种相对独立的功能，某一层不需要知道它的下一层是如何实现的，仅需要知道下一层通过层间接口所提供的服务。表 8-1 给出了 OSI-RM 的层次结构和它们的基本功能。在物理层，数据传送的基本单元称为“帧”；在网络层，数据的传送单元是“分组”，或称为“包”；在传输层(Transport Layer)，则为“报文”。当报文较长时，先将其分割成几个分组，再交给网络层传输。由于传输层之上的各层不再关心信息的传输问题，所以对于高层而言就好像是在两个传输层实体之间有一条端到端的通信通路，因此传输层是十分关键的一层。图 8-1 给出了 OSI-RM 各层在终端和网络节点中的相互关系。

表 8-1　OSI 参考模型

层　　次	功　　能
应用层	与用户应用进程的接口
表示层	数据格式的转换
会话层	会话管理与数据传输的同步
传输层	从端到端有效、可靠地经网络透明地传送报文
网络层	分组传送、路由选择和流量控制
数据链路层	在链路上无差错地传送帧
物理层	将比特流送到物理介质上传送

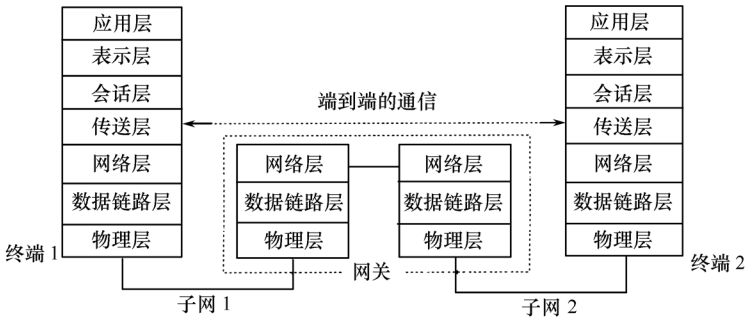


图 8-1　开放系统互连

层次化的协议结构有着易于实现、便于维护、灵活性好、易于标准化等明显的优点。但是，它所采用的下层服务与上层的应用完全屏蔽开的方法，在多媒体通信中存在着一些问题。对于连续媒体来说，传输的实时性是十分重要的，当接收端检测到传输过程中所产生的差错时，一般不要求发送端重新传送丢失的信息，而是根据所丢失信息的内容采取某种补救的措施。例如，对于按 H.263 编码的码流来说，由于采取了帧间编码，某一个宏块数据的丢失不仅导致本帧，还将导致后

续帧相关位置上图像质量的恶化。一种补救的措施是在本帧中用相邻宏块的数据代替(或内插出)丢失的数据,同时将丢失宏块的编号回传给发送端,使编码器在处理后一帧图像时,对相应位置采用帧内编码,以避免误差的传播(见 9.4.4 节)等。对于离散媒体的要求则不同,例如某些控制信号,实时性要求不高,而对准确性要求高,当出现传输差错时要求发端重传往往是必需的。由上看出,在多媒体通信中,对数据传输过程中产生的差错如何处理是和应用密切相关的。但是在层次化协议结构中,下层的服务不了解上层的应用。传统的传输层协议,例如 TCP,对于上层应用提交给 TCP 服务的数据流,TCP 实体仅根据协议,而不是根据应用需要来划分数据单元,得到所谓的**协议数据单元 PDU**(Protocol Data Unit)。由于 TCP 的 PDU 与上层应用所提交的数据的特征(如图像帧、声音样值等)之间没有明确的关系,因而很难针对不同的应用对 PDU 进行不同的处理。

要解决上述问题就需要由上层应用来完成对数据流的划分,这样得到的数据单元称为**应用数据单元 ADU**(Application Data Unit),每个 ADU 相对于原始数据流有明确的含义,例如一个 ADU 对应于一帧图像,或若干个声音样值等。传输层协议将 ADU 作为一个整体来处理,即 ADU 可以进一步被分为若干 PDU,但某个 PDU 发生差错时,传输层协议应能判断该 PDU 属于哪个 ADU,以便根据应用作适当的处理。这就是**应用层分帧 ALF**(Application Layer Framing)的思想[1]。

传统的层次化协议结构的另一个特点是,每一个数据单元必须被逐层处理。这种次序的约束在一定程度上妨碍了协议执行的效率。**集成层次处理 ILP**(Integrated Layer Processing)是另一种协议设计原则[1],其基本思想是在可能的情况下,将对数据的串行操作转换成并行操作,以充分发挥并行处理器、或多任务操作系统的能力,提高协议的执行效率。例如前面提到的报文丢失问题,可以由传输层通知上层发生差错的是哪一个 ADU,由上层作相应的处理;又如对于报文次序颠倒的问题,传统的方式是首先由传输层实体解决报文的排序,再交给应用层作下一步处理。而在某些情况下,例如对于 M-JPEG 编码的码流,由于没有采用帧间编码,其解码过程较少受到报文到达次序的影响,ADU 的排序与解码完全可以并行地处理,等等。

总之,ALF 和 ILP 原则阐明了数据的传输服务应面向于具体应用这一现代通信协议的设计思想,对于多媒体信息的传输尤其有着重要的意义。

8.1.2 传统的因特网传输层协议

我们已在第 7 章中看到,因特网(IP 网)近年来在底层(OSI-RM 的下三层)技术的发展使其性能有了很大的改善,它已逐渐成为多媒体传输的统一平台。本节将讨论在此平台之上的传输层协议。首先在本节中讨论 IP 网传统的传输层协议;在 8.1.3 节中介绍实时传输层协议 RTP,这是根据 ALF 原则专门为连续媒体的传输而提出的;8.1.4 节则介绍由于 IP 网发展的需要出现的新的传输层协议 SCTP。

传统的因特网协议栈中有两类传输层协议,即 TCP 和 UDP。

1. TCP

TCP(Transmission Control Protocol)为在两个进程之间的全双工比特流交换提供可靠的串行通信通道,或称虚电路。由于通过 IP 地址只能识别主机,而每个主机上可能有多个应用程序(或进程)在运行。因此,当两个主机上的某一对进程之间需要进行通信时,每个进程需要向各自的操作系统申请一个**端口**(port),以标明它们之间的连接关系。通过两个主机的 IP 地址和端口号,这一对进程之间就建立了一个 TCP **连接**(虚电路)。一个进程可以有若干个全双工的 TCP 连接。

尽管下层提供的报文服务不是无差错的,TCP 连接却必须保证可靠的、次序不颠倒的比特流传输。为此,TCP 规定收端在接收到每一个包后返回一个**肯定确认**(Positive Acknowledgment)信号。如果发端在一定时间(大于双程往返时间)内收不到确认信号(称为**超时**),则自动从未收到确认的包开始重传。

为了有效地进行传输和进行流量控制，TCP 采用了滑动窗口的机制。窗口内的数据可以连续发送而无须等待确认，但窗口的移动必须等待确认信号到达后才能进行；接收方告知发送方期望收到的数据字节的最小序列号（确认号），这就隐含地告诉了对方，比确认号少一号的数据字节已经收妥，此时窗口可以滑动到确认号处。图 8-2 给出一个数据字节顺序传送的例子。当收到确认信号 ACK 后，窗口移动到图示位置，从而第 17 个字节可以开始传送。显然，当窗口的大小足够覆盖在双程往返时间内可传送的字节数时，传输效率最高。

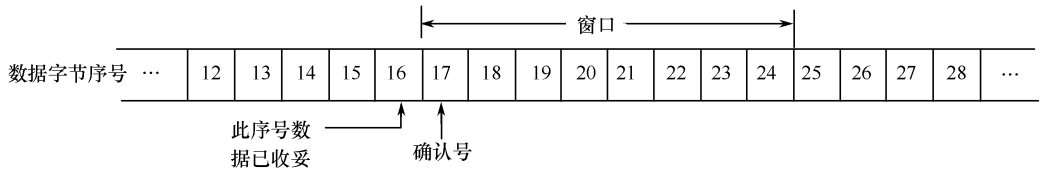


图 8-2　确认号与窗口的含义

所谓**流量控制**是指使速率高的发端所发送的数据流不超出速率低的接收端的接收能力的控制过程。TCP 的流量控制也使用滑动窗口机制，不过窗口的尺寸是可变的。每个从接收端返回的确认信号都包括已接收到的字节数和一个窗口通告（Window Advertisement）。窗口通告给出接收端缓存器还剩余多少空间，根据这个信息发送端改变窗口的大小。若接收端缓存余量大，则增大窗口；反之，则减小窗口。

TCP 还具有**拥塞控制**功能，我们将在 9.2.1 节中具体介绍。

对于多媒体信息传输而言，尽量减少传输过程的延时是十分重要的，定期地传送肯定确认信号会增加数据传输的延时，特别是在长距离传输的情况下影响更大。一般地讲，采用**否定确认**（Negative Acknowledgment）要好一些。同时，重传机制不利于保持音频和视频数据流的连续性。此外，多媒体应用中也不一定总是需要全双工的连接，例如在视频点播系统中，控制信号需要双向连接，而视频信号只需要简单的单向通道即可。最后，TCP 不支持多播。这些都是 TCP 不利于多媒体信息传输的因素，TCP 适用于要求可靠传输的非实时数据。不过在网络带宽充足（重传较少）和接收端容忍一定的起始延时以缓存足够数据的情况下，TCP 也可以用来传输实时媒体数据。

2. UDP

UDP（User Datagram Protocol）建立在 IP 之上，它与 IP 一样提供无连接的数据报文传输。UDP 的功能是将在主机之间交换的数据加上简单的头信息形成 UDP 报文，并检查数据校验和（Checksum）；然后再将 UDP 报文封装到 IP 报文的数据区。报文丢失、次序颠倒、流量控制和拥塞控制等都需要在高层协议中解决。UDP 的简单和高效率使得其实时性比 TCP 要好，且支持多播和广播，但它没有传输层面向连接的功能，这对于提供多媒体信息传输所需要的 QoS 保障是不利的。在实际应用中另一个值得注意的问题是，一些防火墙会阻隔 UDP 报文以防止黑客通过 UDP 注入恶意数据。因此使用 UDP 传输视频数据的多媒体会议、视频点播及 IPTV 等系统均需要注意穿越防火墙的问题。

8.1.3　实时传输层协议 RTP 及 RTCP

实时传输层协议（Real-Time Transport Protocol，RTP）是由 IETF 的音频/视频传输工作组设计的实时传输协议，并已被包括国际电联在内的其他国际标准化组织所接受，成为 IP 网多媒体系统广泛采用的实时媒体传输层协议。

实时传输层协议（RFC 3550）定义了 RTP 的数据报文格式和使用规则，它主要用来为实时数据的应用提供点到点或多点通信的传输服务。伴随 RTP 的还有一个控制协议，称为 RTCP（Real-

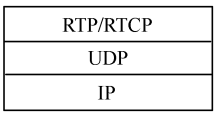


图 8-3　RTP/RTCP 栈

Time Transport Control Protocol)。RTP/RTCP 较好地体现了 ALF 原则。图 8-3 给出了 RTP/RTCP 在因特网协议栈中的位置。RTP 通常运行于 UDP 之上,也可运行于 TCP 之上。

1. RTP

两个、或者多个用户之间建立的 RTP 连接称为 **RTP 会话**(Session)。对于一个参加者而言,会话由一对传输层地址(一个网络层地址加上两个端口地址;一个端口为 RTP 报文的发送/接收所使用,另一个端口为 RTCP 报文的发送/接收所使用)所标识。在一个 RTP 会话上通常只传送一种媒体类型的数据,因而同时传送的音频和视频一般由各自的 RTP 会话连接。RTP 会话相当于我们在第 7 章中提到的逻辑通道,用户数据被封装在 RTP 报文中传输。

为了实现根据应用进行分帧的原则,RTP 定义了两类文件,即格式文件(Format Document)和轮廓文件(Profile Document)。格式文件规定了将某种媒体流划分成 ADU 的原则,以及 ADU 的格式。RTP 已经为 H.26x 视频流、MPEG 视频流以及各种编码格式的音频流等制定了格式文件,例如 H.264 码流的格式文件为 RFC3984。用户也可以根据自己的需要定义新的格式文件。轮廓文件则规定了某类应用对 RTP 的具体使用方法。

图 8-4 给出了 RTP 报文的格式,它包括固定长度的头和载荷两部分。RTP 头中的 V 记录版本号,P 表示该包中有无填充字节,X 表示在固定头之后有无附加的扩展头。**负载类型域**(PT)说明载荷的数据类型和编码方式,在 RTP 的默认轮廓文件(RFC 3551)中,对一些常见的音、视频编码规定了 PT 的具体取值。不同的 PT 值对应于不同的格式文件,也就对应着不同的划分 ADU 的方法。由于每个 RTP 包都包含 PT 域,因此在通信过程中允许改变编码方式,这在有些情况下,如因网络可利用带宽发生变化需要动态地改用对带宽要求较少的编码方式时,是十分有用的。

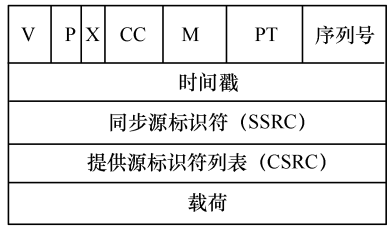


图 8-4　RTP 报文格式

对 RTP 报头中**标记域**(M)协议并没有硬性的规定,但通常用来指示连续码流中的某些特殊位置,例如视频帧的边界、音频码流中非静默期的开始等。**时间戳域**给出与该报文相关的定时关系,通常是指该包第一个字节的取样时间,也可以像 MPEG/RTP(RFC 2038)规定的那样,与 MPEG 码流的显示时间戳 PTS(见 8.2.2 节)有相同的含义等。对于某一个应用而言,以上域具体表示哪种含义,在构成该应用的各个实体所共同约定的轮廓文件中定义。时间戳为收端恢复原码流(在发端)的定时关系提供了依据;同时也为不同码流,例如音频和视频码流之间的同步提供了依据。在有些应用中,例如视频会议,音频和视频码流分别通过两个独立的 RTP 会话传送,以便处理能力不同的终端选择只接收音频、或同时接收音频和视频。这时音频和视频为两个独立的 RTP 流,它们在 UDP/IP 端口上复用。显然,此时发端或收端需要根据时间戳进行流间同步。

如图 8-3 所示,RTP 通常使用不可靠的 UDP 传输服务,包丢失和包次序颠倒是可能发生的。RTP 包头中的 **序列号域**可以用来检测上述问题。值得注意,在有些情况下,例如一个视频帧被分成几个 RTP 包,而时间戳又采用显示时间戳 PTS 时,这些 RTP 包具有同样的时间戳,利用序列号来确定它们丢失与否和它们之间的次序就显得更为重要了。

RTP 支持多方多媒体会议,会议中产生数据的与会者(发言人)称为提供源 **CSRC**(Contributing Source)。多个 CSRC 产生的数据可能通过一个混合设备(如 7.4.2 节讲的多点控制器)复接或合成为一个流进行传输。为了让接收者了解合成流的来源,复合流 RTP 包头中给出了构成该复合流的各个 CSRC 的标识符(通常是其 IP 地址)和 CSRC 的总个数(CC 域)。同步源 **SSRC**(Synchronization Source)标识符则标明负责发送 RTP 包和设置 RTP 包时间戳及序列号的实体。当 RTP 包来自混合器时,该域为混合器的标识符,而不是任何一个提供源的标识符。

值得注意，RTP 与 RSVP 不同，它并不对媒体流在网络中传输的实时性提供任何保障，它只是一个适合于实时媒体传输的传输层协议。它一般不使用 TCP，因而不存在 TCP 重传延时大、不支持多播等缺点。它在 UDP 之上增加了面向连接、可检测包丢失和次序颠倒等功能，弥补了 UDP 不可靠传输服务的不足。同时，RTP 符合 ALF 原则，其包头中含有能够表示数据特点的信息（M 域），使得应用程序无须解码就可以找到码流中的某些特殊点（如帧或宏块的边界），以便于处理。

2. **RTCP**

RTCP 是一个控制协议，它的报文不携带用户数据，只携带与会话有关的控制信息。RTCP 的主要功能，一是发送端通过接收端周期性地反馈回来的 RTCP 报文监测通信质量，如包丢失率、延时及延时抖动等；二是在多方会话中，与会各方通过交换 RTCP 报文获得系统时钟信息、监测通信质量、了解参加会话的用户数，以及实现简单的会话控制功能（如参加/退出会话、识别参加者的身份等）。

RTCP 报文总共有发送者报告、接收者报告、源描述、再见和应用相关功能 5 种类型，其中最重要的是**发送者报告 SR**（Sender Report）和**接收者报告 RR**（Receiver Report）两种。图 8-5给出了 SR 报文的格式，SR 是一个复合报文，除了发送者自己的信息外，还包含它接收其他会话参与者的消息的情况。图中 V、P、PT 和 SSRC 域与 RTP 报文头中的定义相似；RC 域为 SR 中包含接收者报告块的个数；长度域给出 SR 的总长。



图 8-5　RTCP SR 报文格式

SR 的发送者报告部分给出了有关发送时间和发送数据量的信息。其中 NTP 时间戳给出发送当前报文的绝对时间，即按网络时间协议 NTP 计量的时间，或者按会议发起者给出的系统时钟计量的时间；RTP 时间戳的定义方式与 RTP 包头相同。发送者报文计数和发送者字节计数则分别表示发送者从会话开始到当前报文发送时，已经发送的 RTP 报文数和字节数。

SR 的接收者报告部分用于向其他会话参与者报告接收它们发送的 RTP 报文的情况。SSRC_i 是第 i 个接收者报告块对应的会话参与者的标识符。包丢失率域表示从上一个 RTCP 报告以来的 RTP包丢失率，它等于这段时间内丢失的包数除以应收到的 RTP 包数；丢失包总数域表示自 RTP 会话开始以来累计丢失的 RTP 包数；扩展的最高序号域的低 16 位表示接收到的最后一个 RTP 包的序列号，高 16 位表示序号循环的次数（序号超过 65 535 后置零为一个循环）。到达间隔抖动域给出延时抖动的估计值，估计方法将在下面介绍；最近发送者报告时间戳 LSR（Last SR timestamp）域为接收到的 SSRC_i 的最后一个发送者报告中的 NTP 时间戳；SR 最新间隔 DLSR（Delay Since Last SR）域为接收到 SSRC_i 的最后一个发送者报告的时间到本接收报告发送时间之间的间隔。

RR 与 SR 的格式类似，只是它只反映接收其他会话参与者信息的情况，没有自身作为发送者的发送者报告部分。

在举行大型会议的环境下，众多参与者之间需要相互交换 RTCP 报文，这将显著增加网络的负担。为了降低 RTCP 对带宽的占用，协议规定 RTCP 报文的发送间隔不得小于 5 秒，实际的发送间隔通常选为[0.5,1.5]内均匀分布的一个随机数与 5 秒相乘之积。乘以随机数的目的是避免多个参与者同时发送 RTCP 报文。协议还规定 RTCP 报文占用的带宽应小于会话带宽的 5%；所有参与者占用的总带宽不大于会话带宽的 25%。

3. 双程延时和延时抖动的估计

利用 RTCP 携带的信息可以计算收、发两端的双程延时和延时抖动。假设如图 8-6 所示，A 站在时间 T_1 发送了 RTCP SR 报文；B 站在时间 T_2 接收到该报文，随后在时间 T_3 发送了 RTCP RR 报文；A 站在时间 T_4 接收到 B 站发送的报文，那么 AB 之间的**双程延时** T 为：

$$T = T_4 - T_3 + T_2 - T_1 = T_4 - \text{DLSR} - \text{LSR} \tag{8-1}$$

上式中的 T_1 就是 RR 报文中的 LSR，而($T_3 - T_2$)即为 RR 中的 DLSR。

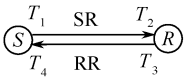


图 8-6　双程延时的估计

相邻两个(第 i 个和第 $i-1$ 个)RTP 报文的传输延时(到达间隔)差 D 为

$$D(i,i-1) = (R_i - S_i) - (R_{i-1} - S_{i-1}) \tag{8-2}$$

其中 R 和 S 分别为对应的 RTP 报文的接收和发送时间，R 由收端在接收到该报文时的本地时间决定，而 S 由该 RTP 报文的时间戳域给出。

报文传输延时差的平均值定义为**延时抖动** J，它可以通过下式估计。

$$J(i) = \alpha J(i-1) + (1-\alpha)\,|\,D(i,i-1)\,| \qquad 0 \leqslant \alpha \leqslant 1 \tag{8-3}$$

上式与(6-11)式一样采用了几何加权平滑函数，我们已经知道，这相当于 $J(i)$ 为 $D(i,i-1)$ 经过一阶低通滤波后得到的结果。在接收报告到达间隔抖动域中返回的就是当前抖动的估计值 $J(i)$。

8.1.4　流控制传输协议(SCTP)

近年来在计算机、通信和广播网络的融合过程中，IP 技术逐渐得到各界的广泛认同，原来在学术和研究环境中诞生的因特网迅速地走向了商业应用，在此过程中逐渐暴露出原有的传输层协议(TCP、UDP)难以满足商业应用的某些需求。例如，当通过 IP 网传送电话的信令消息时，要达到“电信级”的质量，信令消息必须满足严格的时间和可靠性要求。TCP 虽然提供可靠传输服务，但它的重传和保证包正确次序的机制可能引入不可容忍的延时；而 UDP 提供的则是不可靠的传输服务。因此，定义新的传输层协议是必要的，**流控制传输协议**(Stream Control Transmission Protocol，SCTP)就是 IETF 近年来在这方面工作的成果(见 RFC 4960)。

SCTP 像 TCP 一样位于 IP 之上，是一个提供可靠传输服务的面向连接的协议。在传输用户数据之前，两个 SCTP 端点之间必须通过一系列的状态信息交换建立起相互通信的关系，称为**SCTP 联结**(Association)；在数据传输结束后，要拆连(联)。

SCTP 与 TCP 的最大不同之处在于，一个 SCTP 联结中允许有多个单方向**流**(Stream)传输，即允许有多个逻辑的连接，这也是它的名称的由来。每个流用一个流标识符识别。图 8-7 给出了一个具有三个从 A 到 B 和 1 个从 B 到 A 流的联结的示意图。值得注意，既然多个流存在于一个 SCTP 联结中，说明它们共用一对 SCTP 传输层地址。在建立联结时，两个端点协商好本次通信的流标识符的范围(SCTP 协议最多允许 65536 个单方向的流)；联结建立好之后，由应用程序决定使用哪一个流来传递自己的哪些数据。构成同一流的数据包具有相同的流标识符。

SCTP 中引入“流”的概念，为解决 TCP 重传和包顺序传输增加延时的问题提供了灵活的工具。在 SCTP 中，重传和包排序只在“流”的范围内进行。例如在图 8-7 中，假设端点 A 要送两条消

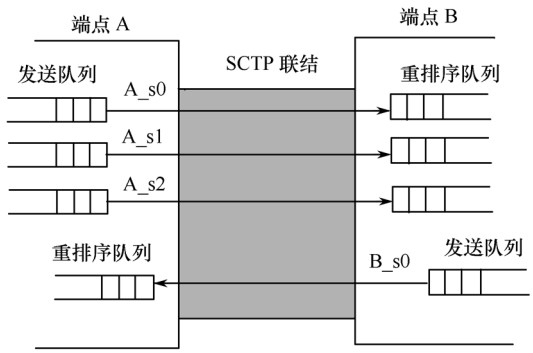


图 8-7　使用 SCTP 流传输多条消息

息到 B，这两条消息分为三段传送，其中第一段包含了完整的第一条消息，但是在线路中丢失；携带第二条消息的第二、三段正确按序到达 B。如果采用 TCP 完成上述任务，尽管收端已经正确收到第二条消息，仍必须等待正确收到重传的第一段之后才能将两条消息输出给应用程序。如果采用 SCTP 完成上述任务，我们可以定义第一段为流 1，第二、三段为流 2。流 2 的数据（第二、三段）正确和按序收到后就可以输出，而不必关心在它前面的流 1 的数据是否正确到达。

支持**多址工作**（Multi-homing）是 SCTP 的另一个关键特征。我们以图 8-8 为例，假设过程 1 和 3 之间存在一个多址 SCTP 联结（网卡 IP1、2 和 IP 3、4）；过程 2 和 4 之间存在一个 TCP 连接（IP2 到 IP4）。如果网卡 IP 2 损坏，过程 2 和 4 之间的 TCP 连接不可恢复，但过程 1 和 3 之间由于还存在 IP1 和 IP3 之间的通路，因此通信不会中断，甚至应用程序（如果它选择如此的话）可以不知道已经发生的部件损坏。

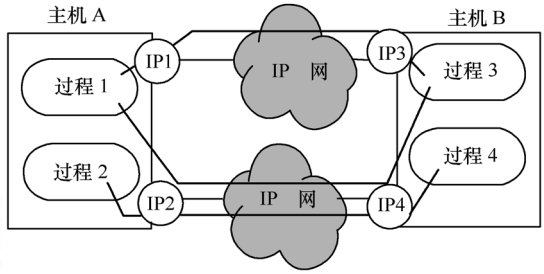


图 8-8　多址和单址工作的比较

如前所述，SCTP 的初始设计目标是在 IP 网上支持电信级的信令消息传输，多媒体系统的信令消息，如 SIP（见 8.4 节）和 RTSP（见 8.6.3 节）等，可以在 SCTP 之上传送。同时，由于 SCTP 支持多址和多流工作、具有和 TCP 相同的拥塞控制等优点，特别是 IETF 将 SCTP 作了扩展，在 RFC 3758 中定义了**部分可靠的 SCTP**（SCTP Partial reliability extension），研究人员开始尝试用 SCTP 代替 UDP 应用于实时媒体的传输，例如 IPTV、VoIP 和 10.4 节中介绍的多描述码流等，期望获得更好的 QoS 和提高网络资源的利用率[3]。

8.2　多媒体数据的复接与分接

在存储或传输之前，多种不同媒体流通常需要复接在一起形成一个单一的流。复接的方法主要有时分复用和分组复用两种。

8.2.1　同步时分复用

本节将以国际电联 H.221 标准为例介绍同步时分复用的原理。

H.221 是国际电联制定的同步电路交换网上 H.320 视听会议终端中使用的复接标准(见 8.3.2 节)。在此标准中,每 80 个字节(10 ms)的数据构成一帧。每个字节的对应比特构成一个 8 kHz的子信道,例如所有字节的第 1 比特构成第 1 个子信道(见图 8-9)。第 8 个子信道称为**公务子信道 SC**(Service Channel)。在 SC 中,前 8 个比特作为**帧校准信号 FAS**(Frame Alignment Signal)。FAS 给出帧的校准(同步)信息、告警信号以及错误检测等。SC 的第二个 8 比特称为**比特率分配信号 BAS**(Bit-Rate Allocation Signal)。BAS 实际上是通信控制信号,用来描述终端具备的能力、信道的构成(当采用多个 B 信道时)、控制与指示信号等。SC 的第三个 8 比特作为**加密控制信号 ECS**(Encryption Control Signal)。SC 的剩余带宽可以由用户数据所占用。

图 8-9 所示为 N-ISDN 基本接口的两个 B 信道(2×64 kb/s)共同使用时子信道的占用示例。除了公务子信道之外,第一个 B 信道的第 0～5 个子信道传输音频数据,其余子信道传输视频数据,图中 A_1、A_2…和 V_1、V_2…分别代表音频和视频数据的第 1、2、…、n 个比特。从图上可以看出,这实际上是一种**同步**时分复用的方法。不同媒体的数据分别在所分配的固定的时隙中传送,因而它们各自占有的带宽是固定的。当需要传送用户数据时,数据也可以占用某一个或几个子信道。如果某种媒体的速率低于分配给它的子信道的速率,剩余的带宽不能为其他媒体所使用。

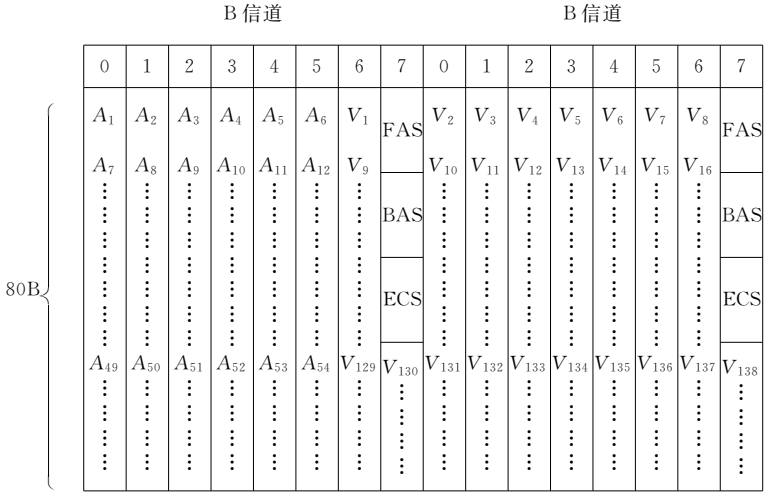


图 8-9　两个 B 信道构成的帧结构

依照与图 8-9 相类似的方法,相邻的 $n\times64$ kb/s 信道可以组合以获得更高的信道容量,如 H0、H12/H11 等。在接收端,解复用器首先找到一帧的开始,然后按 BAS 给出的信息将复接的码流分离成单一媒体的数据流。

8.2.2　分组复用

分组复用、或称**包复用**,将不同的媒体流分别按一定的长度(变长或固定长度)分组,每一组码的前面加上包头,然后在同一信道上轮流传送不同媒体类型的包。这可以看成是一种**异步**的时分复用。在接收端根据包头信息将各种包区分开来,去掉包头重新组合成各自的码流。异步时分复用的带宽分配比较灵活。

本节将以国际电联的 H.222.0/H.222.1 标准作为例子介绍分组复用的原理。H.222.0 实际上是 MPEG 2 的系统层(ISO/IEC 13818-1)协议,它在视频压缩、存储、传输,以及数字电视广播和 HDTV 中得到广泛应用。

1. 节目流和传送流的复接/分接

将不同媒体,例如视频和音频的**基本比特流**(Elementary Bit Stream)复接的方式有两种[见

图 8-10(a)]。首先都是将基本比特流按应用的需要打成长度不同的包，形成**包基本码流 PES**(Packet Elementary Stream)。第 1 种复接方式将 1 个或多个 PES 包组合，并加上包头构成**大包**(pack)，然后将大包复接形成**节目流 PS**(Program Stream)。需要注意，这里复接的 PES 流必须是来自同一节目的(如视频及其伴音)，即它们有共同的时间基准(时钟)。另一种方式是将 PES 包连同 PES 的包头都看成是数据，将这些数据分成等长度的段，重新打成长度固定的包，并加上包头，称为**传送包**(Transport Packet)。然后轮流传送传送包，形成**传送流 TS**(Transport Stream)。一般规定，PES 包的始端要放在传送包的起始处。因此，如果上一个传送包还未填满的话，则要插进一些冗余比特将其填满。图(c)给出了一个等长传送包复接的数据流的示例。

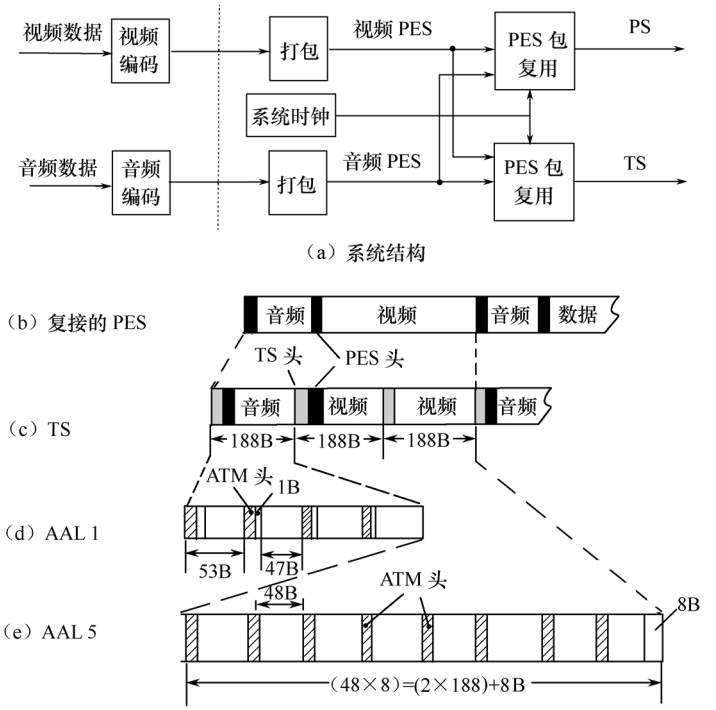


图 8-10　两种复接方式

除了视频和伴音信号以外，其他附加数据、控制数据等也都可以在经过打包之后复接到同一个信道上。由于传送包头中含有**包标识符 PID**(Packet Identification)，用于标识该包属于哪一个流，因此所复接数据的类型不必事先限制，只要给以适当的 PID 就可随时插入。例如，在传输一个需要付费的节目之前，可以先插入密钥的传送。此外，如果在将来出现新型业务时，新业务的基本码流也可以很容易地插入而无须作硬件方面的改动。对于接收机来说，不管复接了多少种码流，只要在固定位置上找到包头中的 PID，就可以将每一种码流的包一一分离出来。对于节目流来说，PES 包头中有**流标识符 SID**(Stream ID)，与 PID 的作用类似，借助于 SID 很容易实现不同流的插入和分离。

直接采用 PES 包的复接方式比较简单，但在信道产生误码或失去同步的情况下，由于包长不固定，检错、纠错和再同步处理都比较复杂，因此这种方式适用于在几乎无误码的环境(如 VCD 或 DVD)中使用。而传送包复接方式比较适于在通信时存在误码的情况。传送包是等长的，在检测到误码时，因为已知包的长度，从下一个无误码包开始就可以使数据流的接收恢复正常。同时，在等长包上容易进行信道编码，例如在卫星、电缆和地面数字电视中，都是在传送包上进行纠错编码(RS 码、卷积码)的。

PES 包的长度可随业务类型的不同而不同，其允许的最大长度为 2^{16} 个字节。集合了几个 PES 包的大包的长度没有具体规定，因为节目流旨在无误码环境中应用，因此一般包长较长，包头的开

销较小。大包的包头中包含有该节目的定时信息、比特率等。有关系统的参数由系统头携带，系统头在节目流开始时传送，同时可以在节目流的其他部分重复传送。

传送包的长度规定为 188 个字节。由于多媒体业务原来期望在 ATM 网上传送，因此包的长度与 ATM 信元长度相协调。如果使用 AAL1(固定速率 ATM 适配层)，一个 ATM 信元，除包头以外再用一个字节传送 AAL1 头信息，剩余 47 个字节传送数据。这样 4 个信元恰好传送一个 H.222.0 的传送包[见图 8-10(d)]。如果使用 AAL 5，将两个 188 字节的传送包一起交给 AAL，加上附加在 AAL 5 PDU(Protocol Data Unit)后面的 8 字节的尾信息，则正好构成 8 个信元[见图 8-10(e)]。

传送包的包头中包含包同步的标志、PID、优先级和加扰控制指示等。在有些情况下，节目必须**有条件地接收**(Conditional Access)，例如收费节目只有交费后才能收看。为此发送端将传送包中的有效数据加扰。所谓**加扰**，是将数据流按某种规律变成伪随机序列。解扰时利用传送包头中的一个专门区域(或另一个专门的 PES 流)，将解扰的密钥传给收端，使已交费的收端能将加扰的数据恢复成正常信号。解扰的密钥还可以随时间不断变换。传送包头中的加扰控制指示用来标志该包中的数据(包头信息永远不加扰)是否加扰；如果已加扰，应该用哪一种密钥来解扰。

传送包的包头中除固定长度部分外，还有一个变长部分，称为**适配包头**，它的主要功能如下：

(1) 模拟电视信号的行、场时间间隔是固定的，接收机可以从全电视信号中提取出同步信号(见图 2-14)。而在已压缩的视频码流中，每幅图像产生的数据量是不相同的，没有与模拟电视中的同步脉冲相对应的信息。为了解决这一问题，每隔一段时间，在适配头中传送一段27 MHz的时钟脉冲 **PCR**(Program Clock Reference)给接收机作为时间基准，用来调整接收机本地时钟的频率与相位，使之与发端同步(见 6.6.2 节)；

(2) 当用户开机或转换频道时，接收机的解码器是从节目的中途开始工作的。我们知道，只有 I 帧图像不需要参照前面的信息来解码，因此只有从 I 帧开始，解码器才能正常工作。我们将 I 帧的起始点称为**随机接入点**(Random Access Point)。在适配头中有一个标志位，标志着该传送包中是否含有随机接入点；

(3) 在节目播送过程中，有可能需要插入一段本地节目，例如地方台可能在中央台节目中插入本地的广告。与频道转换类似，本地节目必须在随机接入点插入，同时还必须保证插入信号不引起解码器缓存器的变空或溢出。本地节目以传送包的形式插入原节目流。在适配头中有一个倒计数标志，在节目切换前的几个包开始计数，当计到 0 时是原节目的最后一个包，计数到－1 则是另一个节目的开始。此外，解码器在节目切换前后接收到的两个包的 PCR 属于两个节目源，两者的时钟是各自独立的。如果这时解码器时钟盲目地继续以锁相方式来跟踪的话，很可能完全失去同步。为了防止这种情况，在切换后的第一个传送包的适配头中，用改变某个标志位的办法告诉解码器 PCR 已经更换，应当直接改变时钟的相位；

(4) 除以上功能外，适配头还可以扩大，以支持某些专门的或者新的功能。

2. 单节目和多节目传送流的复接/分接

传送流的复接分两个层次进行。在底层，将所有共用同一时间基准(即使用同一时钟)的基本码流(例如视频和伴音的 PES)复接成一个节目传送流。在高层，使用不同时钟的节目传送流(例如不同电视台的节目流)再复接汇合成一个多节目 TS 流，或称为**系统码流**。图 8-11(a)给出一个底层复接的例子。除了基本码流之外，还有一个控制码流。控制码流传送一个**节目映射表** PMT (Program Map Table)，用来说明组成这个节目的基本码流的 PID 号，以及这些基本码流之间的关系等。图 8-11(b)是高层(也称作系统层)复接的例子。除了节目传送流之外，还有一个 PID ＝ 0 的系统控制流，用来传送**节目关联表 PAT**(Program Association Table)，该表给出携带不同的节目映射表的节目控制码流的 PID。要分离出某个节目的码流，首先从节目关联表中查到节目映射表

的 PID，再从节目映射表中找到构成该节目的所有基本码流的 PID，分接设备就可以根据查到的 PID 将所需要的节目流分离出来。图 8-11(c)是上述分接过程的示意图。

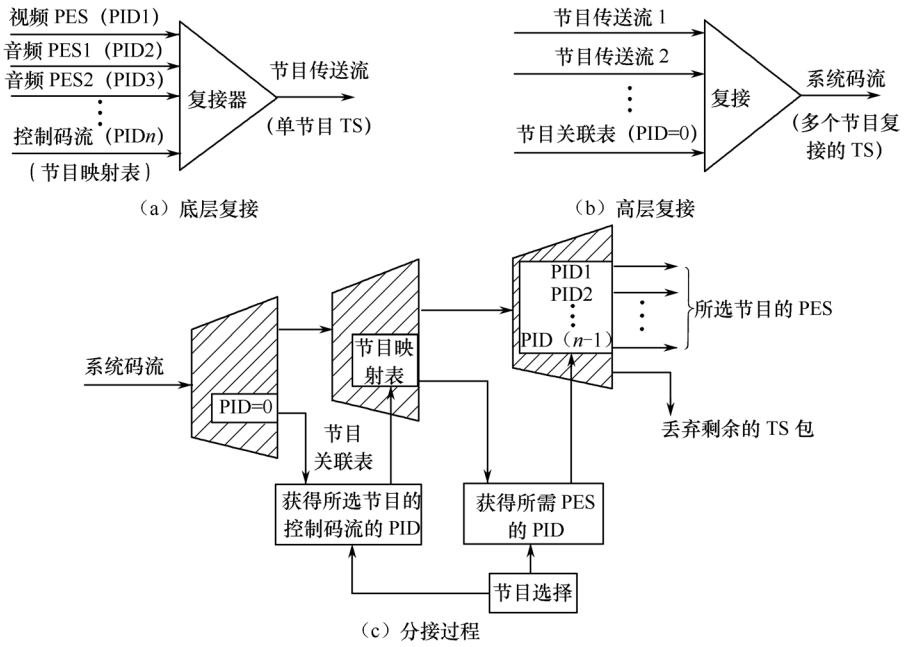


图 8-11　TS 流的复接/分接过程

顺便指出，在传送流中除了 PAT 和 PMT 之外，还可能包含有网络信息表 **NIT**(Network Information Table)和条件接收表 **CAT**(Conditional Access Table)。PAT、PMT、NIT 和 CAT 合起来称为**节目特定信息 PSI**(Program-Specific Information)。NIT 是节目的广播者用来传送与网络有关的信息的，CAT 则包含与条件接入或与扰码有关的信息。PSI 作为特殊数据在控制码流 TS 包的载荷中传送。

3. 定时信息的传送

我们知道，收发时钟的同步是保证连续媒体流正确播放的重要条件。为了使接收端能够获得发端时钟的信息，在复接过程中必须在 TS 流(或节目流)中周期性地插入时钟信息——PCR。PCR 在 TS 适配包头的 PCR 域[见图 8-12(b)]中传送，它是系统 27 MHz 时钟的 42 比特计数值，前 33 比特为 27 MHz 经 300 分频后的计数，后 9 比特为余数的计数。接收端根据 PID 选择出所需的节目，抽取出对应的 PCR，然后利用 6.6.2 节介绍的收发时钟同步方法，使收端时钟与发端保持一致。对于节目流而言，定时信息的传送是类似的，只是计数器值由大包包头的 **SCR**(System Clock References)域携带[见图 8-12(a)]。

如第 6 章所讨论的，基准时钟频率应有较高的准确性和稳定性。H.222.0/H.222.1 要求 27 MHz 时钟的频率偏差小于±810 Hz，频率漂移小于 75×10^{-3} Hz/s。对于单节目流，SCR 至少间隔 0.7 秒传送一次；而对于多节目流，PCR 则至少间隔 0.1 秒传送一次，以向接收端提供足够的发端时钟信息。

当同一节目的不同媒体流(如视频与其伴音)复接时，保持媒体流之间的同步是十分重要的。在 H.222.0/H.222.1 中，将节目的 27 MHz 基准时钟经 300 倍分频后得到 90 kHz 时钟，该时钟驱动一个 33 比特计数器，当进行视频或音频 PES 打包时，在包含一帧数据开始的 PES 包头中插入计数器的当前读取值，称为**解码时间戳 DTS**(Decoding time stamp)，表示该帧的解码时

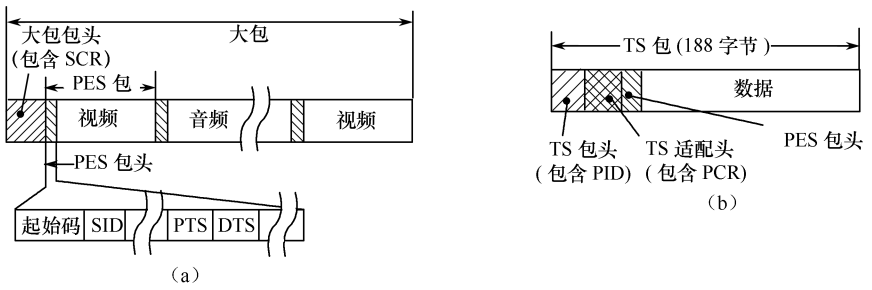


图 8-12　H.222.0/H.222.1 的定时信息(a)节目流;(b)TS 流

刻。由于 MPEG 压缩时存在 I、P、B 帧的差别,其解码次序和显示次序有所不同,因此,在 PES 的包头中还插入了**显示时间戳 PTS**(Presentation Time Stamp),用以表示该帧的显示时刻[见图 8-12(a)]。对于音频流而言,只有 PTS,没有 DTS。显然,要保持流间同步,发端音、视频流的时间戳必须由同一个 27 MHz 时钟驱动的同一计数器产生;而在接收端则应保证本地 27 MHz 时钟与发端时钟同步,并且音、视频包按照相同的时间戳(PTS)同步播放。

4. 复接器的包调度

在进行包复用时,不能像同步时分复用那样对各种媒体进行子信道的预分配,那么在某个时刻复接器应该输出何种媒体的包呢,这就是包调度器应该解决的问题。在这里,我们给出复接器应该满足的一般要求,并以一对音、视流构成的单节目 TS 流为例,对复接中的包调度问题加以说明。图 8-13 给出了这个例子的框图,其中包调度器决定输入码流如何交替输出。复接器除了交替传送音、视频包以外,还要周期性地传送定时(PCR)和服务信息(PSI)等辅助信息。

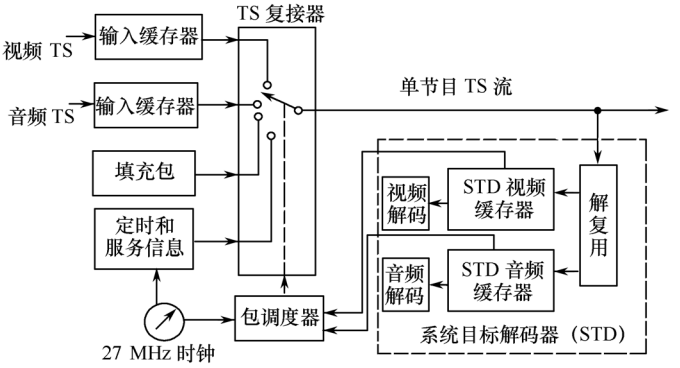


图 8-13　单节目 TS 流复接的例子

一个复接器需要满足下列基本要求:

(1) 复接后的最高速率小于传输通道允许的速率。值得注意,复接后的速率除了由输入码流的码率之和决定外,还必须考虑 TS 包头、PCR 和 PSI 传送等所占用的带宽。根据应用的需要,复接器的输入和输出码流均可能是恒定或变速率的。

(2) 保证复接器输入缓存器不上溢或下溢。各输入码流以自己的速率(可能相同也可能不同)持续向输入缓存器注入数据,复接器必须通过恰当策略进行包调度,以保证各输入缓存器不上溢或下溢。

(3) 保证接收端解码器缓存器不上溢或下溢。以图示的单节目 TS 流复接为例,在解码端音、视频流以正常速率同步播放时,接收端的各个缓存器都不出现上溢或下溢。为了分析的方便,通常在复接端利用一个虚拟的解码器,称为**系统目标解码器 STD**(System Target Decoder)来模拟接收端的操作(每隔一个帧周期从 STD 缓存器中取出一帧数据),为复接器的包调度提供依据。图 8-13 中给出了本例条件下的简化 STD 框图。

在保证 PCR 和服务信息传送的同时，包调度器可以采用的一个策略是：对当前在输入缓存器中等待的视频和音频数据单元到达各自的 STD 缓存器的时间 t_v 和 t_a 进行估计，并计算 $\min\{DTS_v - t_v, DTS_a - t_a\}$，其中 DTS_v 和 DTS_a 分别为视频和音频数据单元的解码时间戳，差值为该数据单元在 STD 缓存器中停留的时间；包调度器选择差值较小的媒体包输出（防止该媒体 STD 缓存器下溢）。同时，如果视频和音频 STD 缓存器均没有再容纳一个数据包的空间时，复接器为防止两个 STD 缓存器的上溢，不输出视、音频包，而输出一个填充包，以保持复接流速率的恒定。填充包在解复用时被丢弃。

以上我们介绍了复接器应该满足的基本要求和一个借助于 STD 缓存器状态进行恒定速率单节目 TS 流复接的包调度的例子。相类似地，基于复接器输入缓存器的状态进行复接器的包调度也是可行的。此外，当输入码流有不同的时间基准（多节目）或输入和输出码流为变速率码流时，要满足复接器的基本要求，其包调度策略会更复杂些。

8.2.3 多媒体数据的封装

从第 7 章我们看到通过分组交换（IP）网络提供多媒体业务已经是技术发展的趋势，这就要求我们将多媒体数据封装成包进行传输。本节将对如何打包进行扼要的讨论。

1. 定长包与变长包

将数据打成定长或变长的包有各自的优缺点。当包长固定时，有以下的优点：

（1）有利于信道容量的分配；

（2）有利于检错和再同步（已知包的长度很容易找到下一个无误码包的开始）；

（3）有利于加纠错码和进行数据交错（见 9.3 节）；

（4）有利于对包进行处理（如交换、复用）的设备的简化。

当包长不固定时，有以下优点：

（1）有利于保持数据的逻辑结构（如一个视频帧数据打成一个包）；

（2）有较高的效率（不需要在应用数据单元不等于包长的整数倍时添加填充字节）。

因此，应该根据不同应用的需要对多媒体数据进行封装，8.2.2 节中介绍的在 DVD 等记录应用中采用变长包（PES 包）、在传输中采用定长包（TS 包）就是很好的例子。

2. 包长的选择

在一个多媒体系统中，数据包长度的选择应该考虑如下因素：

（1）包头所占的比例

应用数据在分段打包时，每一个包要加一个包头，如果包长过短，包头在整个包中占的比例大，每个包所携带的应用数据少，则传输效率降低，或者说打包的代价高。

（2）处理延时

可以理解，包的长度越大，由于中间节点必须将包完整接收下来，才能转发给下一个节点，因此传输延时越长。但是长包可以减少节点查找路由表的次数。

（3）链路带宽利用率

对于同样的数据量来说，长包的包间隔少，因而链路层带宽利用率较高。

（4）网络最大传输单元

许多分组交换的网络有限定的**最大传输单元 MTU**（Maximum Transmission Unit）。如果包长大于目的接入网或中间网络的 MTU，目的地网关或中间路由器将包拆分成几个较小的单元，其长度由途径网络的 MTU 决定，这往往是我们不希望的。下面以在令牌环和以太网上的两个主机间的通信为例来说明这个问题。假设位于令牌环上的主机 A 有一段长为 7000 B 的数据（包括传输

层协议包头)要向以太网上的主机 B 传送。由于令牌环的 MTU＝4000 B,IP 报头长 20 B,一个令牌环帧最大的可用数据长为 3980 B,在分段时所有用户数据段(最后一段除外)的长度须是 8 字节的整数倍,因此用户数据被分为一个 3976 B 和一个 7000－3976＝3024 B 的包传送(见图图 8-14 左侧)。当这两个包到达以太网接入网关时,由于以太网的 MTU＝1500 B,除去 IP 包头 20 B,一个以太网帧的最大可用数据长度为 1480 B(可被 8 整除),从 A 来的每个包要进一步分成三个小包,每个小包的长度见图 8-14 右侧。主机 B 将这 6 个包重组成 7000 B 的数据块,传送到对等的传输层。如果传输层采用的是 TCP 协议,那么这 6 个包中的任何一个丢失,都会导致 7000 B 数据的重传;如果传输层采用的是 RTP/UDP 协议,丢失一个包则可能导致接收到的其余 5 个包也无法利用。而在 IPV6 中,如果包长大于 MTU,路由器不进行拆分直接将包丢弃(见 7.7.3 节)。由此看出,传输层的最大数据分段应该限制在传输网络的 MTU 之内,同时根据 ALF 原则该数据段应该是独立可处理的。例如,如果将一帧视频数据放入一个 RTP 包、包长超过 MTU 的话,更为合理的方案则是在应用层将帧分成条(或 MB),每一个长度不大于 MTU 的 RTP 包包含 1 个或多个条(或 MB)。当传输网络的 MTU 未知时,源 IP 可以利用**互联网控制报文协议 ICMP**(Internet Control Message Protocol)来发现沿途链路的 MTU。

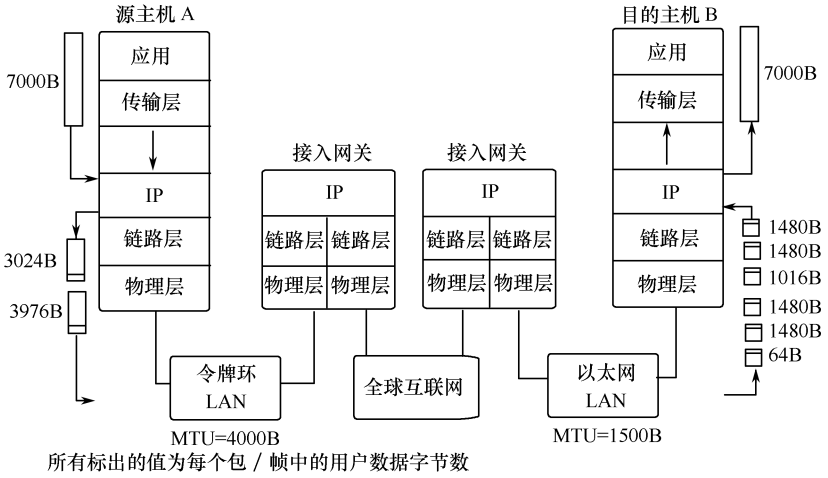


图 8-14　因特网报文的分段和重组

(5) 传输网络的错误率

假设信道产生随机错误,且误码率为 P。当每一个数据包均包含的比特数为 N 时,一个包中至少包含一个错误的概率为

$$P_N = 1-(1-P)^N \approx NP \tag{8-4}$$

当 $NP<1$ 时,右边的近似式成立。上式说明 N 越大,包发生错误的概率就越大。假设发生错误的包必须丢弃(或重传),则包越长,丢失(或重传)的可能性越大。因此,对于误码率较高的信道应该选择较短的包长,例如无线信道的误码率一般为 10^{-3},包长通常选择在 100 B 左右,甚至更短。而在有线网络上,包长则主要依据 MTU 来确定。

(6) 与底层网络协议单元的配合

一个典型的例子是 8.2.2 节中介绍的 TS 包长的选择。TS 包长选择为 188 B,当通过 ATM 网传输时,与视频最常采用的两种 ATM 适配层(AAL1 和 AAL5)的打包方式能很好配合,无须加入任何填充字节,有较高的传输效率。

3. 包头压缩

音、视频数据在进入链路层之前需要经过 RTP/UDP/IP 封装,表 8-2 给出各层封装引入的包

头长度。对于 IPv4 和 IPv6 而言，网络第 3 和 4 层包头总长分别为 40 B 和 60 B。另外，为了保证实时性，多媒体数据的有效载荷一般较小，例如 IP 电话(VoIP)的载荷为 15～20 B。与载荷相比过长的包头使得链路效率大大降低，这对低速链路的影响尤为突出。

当我们仔细地审视这些包头时，不难发现其中存在很大的冗余。对于属于同一包流的包，一些头字段是不变(静态)的，例如源和目的地址、源和目的端口，以及媒体类型等。对于同一包流中的连续的包，一些头字段是线性递增的，例如序列号和时间戳等；只有少数头字段是每个包都变化的。我们将这些情况总结在表 8-3 中。如果对那些在包头中一直保持不变的字段，只在包流开始传送时发送一次；而对那些线性变化的字段采用差分预测编码，则可大幅度地减小包头的开销，这即是**包头压缩**的概念。RTP/UDP/IP 包头一般可压缩到 2～5 B。假设 RTP/UDP/IP v6 包头为 60 B，VoIP 载荷为 20 B，将包头压缩至 2B，链路有效吞吐量可以提高 3.6 倍。

表 8-2　RTP/UDP/IP 包头长度

协　议	包 头 长 度
RTP	12B
UDP	8B
IPv4/v6	20B/40 B

表 8-3　RTP/UDP/IP 头字段属性

	RTP	UDP	IPv6
静态	版本号；P；X；CC；PT；同步源标识符	源端口；目的端口	版本号；业务等级；流标志；下一个包头域；源地址；目的地址
线性变化	序列号；时间戳		段数限制
变化	M	校验和	

包头压缩通常在用户到边缘接入设备的低速率串行链路中进行，在接入设备中包头经过解压缩恢复成标准的 RTP/UDP/IP 包头后进入高速骨干网。如图 8-15 所示，对每一个包流在压缩和解压缩器中都保存着预测编码的参考包头。如果在传输过程中发生包丢失，那么压缩器和解压缩器中的参考包头将不对应(也称为不同步)，解压缩后的包头将发生错误。这个问题在无线链路上更为严重。一个好的包头压缩方案应该提供迅速恢复压缩和解压缩双方参考包头同步的机制。

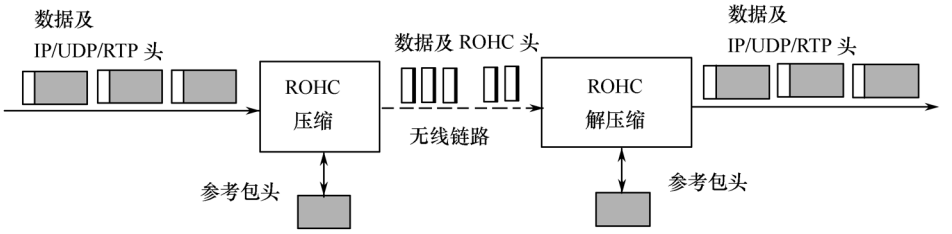


图 8-15　包头压缩方框图

早在 20 世纪 90 年代初 IETF 就制定了包头压缩的协议 RFC 1144，该协议仅针对 TCP/IP 包头压缩，目前已广泛地应用在 IP 协议栈中来提高低速串行链路的效率。此后为适应网络发展的需要，还出现了一些其他的包头压缩协议。近年来由于无线网络和多媒体应用的发展，IETF 相继制定了 RFC 3095 鲁棒包头压缩 **ROHC**(Robust Header Compression)和 RFC 3545 RTP 增强压缩 **ECRTP**(Enhanced Compressed RTP)两个协议。ECRTP 用于具有长延时、包丢失和序号颠倒链路上 RTP/UDP/IP 的包头压缩；ROHC 用于高差错率和长延时链路(例如无线链路)上 RTP/UDP/IP 的包头压缩。ROHC 比 ECRTP 具有更好的抗差错性能，这是以实现的复杂度增高为代价的。目前应用于骨干网的包头压缩协议也在研究之中。

8.3　视听通信系统与终端

8.3.1　视听通信终端的一般框架

人们知道，接入通信网的设备(电话机、传真机、交换机等)必须遵从相应的国际标准，否则不同

国家和不同厂家生产的设备之间将不能互通，也就不能达到通信的目的。对于多媒体通信设备也是如此。本节将介绍国际电联为视听通信系统(Audiovisual Communication System)制定的有关标准，这些标准主要是针对多媒体会议和可视电话业务的，但也不排除其他的应用。

视听业务在两个、或多个用户之间提供图像和伴音的实时通信，其中图像通常是活动图像，但也可以为静止图像、图形或者其他形式。ITU-T 第 15 研究组为在几种主要网络环境下使用的视听系统制定了标准，这些标准都归结在 H 系列建议中。图 8-16 给出了 H 系列视听通信终端的一般框架，其中通信控制信号和应用数据可以和压缩编码后的音频和视频信号复接在一起，也可以直接进入网络的适配层。

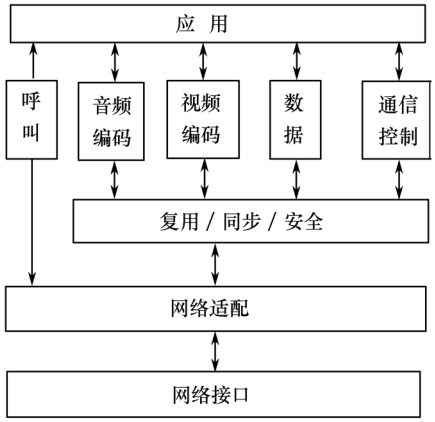


图 8-16　视听终端的一般框架

由于 ATM 技术不再在终端中使用，我们将不介绍 ATM 网视听业务标准 H.310 和 H.321。ITU-T 建议的其余视听业务标准见如下几节。

8.3.2　同步电路交换网视听业务标准(H.320)

H.320 是国际电联为同步电路交换网(如 N-ISDN)进行视听业务制定的标准。H.320 描述整个系统，与之相关的一系列建议则定义构成系统的各个部件。图 8-17 给出了 H.320 终端的协议参考模型，图中呼叫控制采用 N-ISDN 的呼叫信令 Q.931，视频编码采用实时性强的 H.261、H.263 和 H.264 标准，音频编码主要针对语声信号，包括 G.711(未压缩的 PCM 信号)、G.728(16kb/s)，以及高质量语音压缩的 G.722 等标准，复接采用 H.221 标准。数据控制协议 T.120 系列、H.239 和通信控制协议 H.242 则将在 8.3.6 节中介绍。

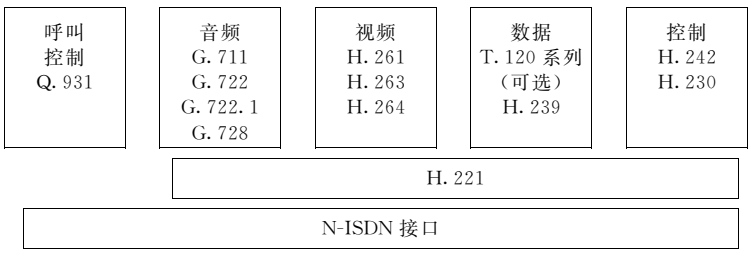


图 8-17　H.320 终端协议参考模型

H.320 系统在 64 kb/s(B 信道)、384 kb/s(H0 信道)和 1536/1920 kb/s(H11/H12 信道)上提供视听业务。这是 ITU 最早批准的标准，在会议室会议电视系统中被广泛应用。我们知道，对于电路交换的网络，需要加入 MCU 设备来支持多点通信。表 8-4 列出了有关 MCU 的协议，例如

H.231、H.243 和 H.230、以及 H.320 系统中的其他有关协议，供读者参考。

表 8-4　H.320 系统的其他有关协议

H.231	2 Mb/s 以下数字信道的视听系统多点控制单元
H.242	2 Mb/s 以下终端之间的通信规程
H.243	多个终端与 MCU 之间的通信规程
H.230	帧同步控制与指示信号
H.233	视听业务的加密系统
H.234	视听业务的密钥管理与认证
H.281	电视会议远端摄像机控制规程
H.228	通过 H.281 和 H.228 实现会议终端间的互控
H.224	同步信道的集合

8.3.3　分组交换网视听业务标准(H.323)

H.323 原来是针对没有 QoS 保障的局域网环境的视听业务而制定的，只要带宽和延时满足要求，它也可以应用到更大范围，如城域网和广域网。1997 年国际电联重新定义了 H.323，使它成为在不保证 QoS 的分组交换网上的标准。在这种环境下，参加视听会议的所有终端、网关和多点控制单元(MCU)，以及对它们进行管理的**网守 GK**(Gate Keeper)的集合，称为一个带(Zone)，或称为**域**(Domain)。一个域中至少有一个以上的终端，可以有、也可以没有网关或 MCU，但必须有一个、也只能有一个 GK。图 8-18 给出了一个 H.323 系统的域。由于 LAN 对接入没有控制，因此引入 GK 对域内的终端进行接纳控制(接纳准则未标准化)以防止拥塞。同时 GK 还具有限制某个终端所使用的带宽、进行地址翻译和域控制等功能。由此看出，GK 的存在对于改善在无 QoS 保障的分组网上的视听业务的质量是有益的。H.323 网关用于 H.323 与其他类型的终端，如 H.320 和 H.324 等之间的连接；MCU 则提供会议管理以及视频、音频信号的混合与切换等功能。我们将在 MCU 中完成音频、视频信号的混合与切换功能的部分称为**多点处理器** MP(Multipoint Processor)，完成其余功能的部分称为**多点控制器** MC(Multipoint Controller)。由于局域网支持多播，因此在 H.323 域中可以没有 MCU，会议由分布在终端、网关或 GK 上的 MC 进行分布式管理；而音视频信号则由每个会议终端利用多播向所有其他终端传送。

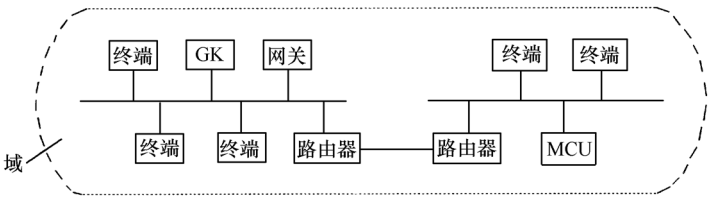


图 8-18　H.323 系统的域

在分组网上的 H.323 终端的协议参考模型如图 8-19 所示。由于缺乏 QoS 保障，H.323 终端使用低码率的音频和视频编码标准，如 G.723.1、G.728 和 H.263、H.264 等。在单纯的 LAN 上使用时，音频还可以采用 G.711，G.722，G.729。音、视频信号采用实时传输协议 RTP，并使用非可靠传输层服务 UDP 进行传输。图中涉及媒体打包和控制的 H.225.0 层定义在传输层(如 TCP/UDP/IP)之上，与 H.225.0 有关的协议栈如图 8-20 示。在进行任何通信之前，H.323 终端必须先找到一个 GK，并在那里注册。用来传输注册、接纳和状态信息的非可靠信道称为 **RAS 信道**。在启动一次通信时，首先通过 RAS 信道向 GK 传送一个接纳请求。当被接纳后，则通过一个可靠信道利用 Q.931 进行呼叫，该可靠信道的传输层地址可能是接纳被接受时返回的，也可能是

呼叫方已知的。呼叫过程结束后建立起一个可靠的 H.245 控制信道。一旦控制信道在收、发端之间建立起来之后，则开始能力交换协商和为音频、视频及应用数据建立子信道。我们知道，连续媒体和离散媒体在传输方面的要求很不相同，不同的媒体采用不同子信道传输，可以对各子信道提出不同的 QoS 要求，从而有效地利用系统和网络的资源。

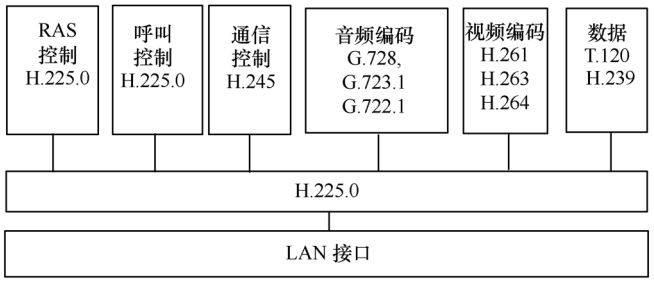


图 8-19　H.323 终端协议参考模型

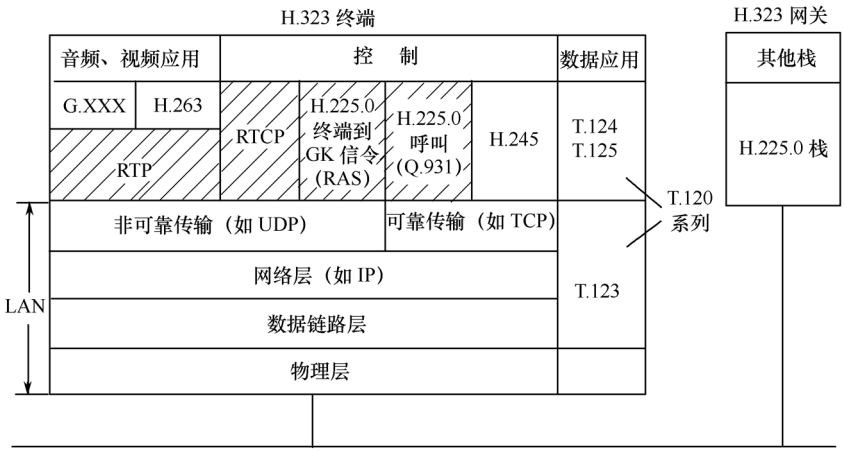


图 8-20　与 H.225.0 相关的协议栈

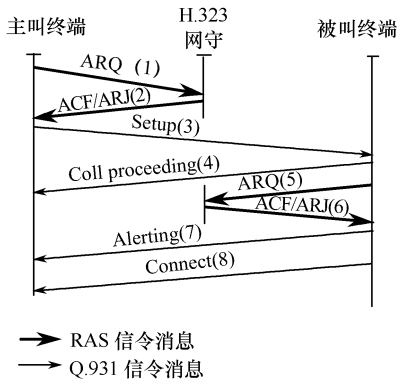


图 8-21　通过网守的呼叫建立过程

H.323 建立呼叫的具体过程如图 8-21 所示。

(1) 首先主叫终端向网守发送接入请求(ARQ)消息；

(2) 网守回应接入确认(ACF)或者接入拒绝(ARJ)消息；

(3) 如果呼叫请求被接受，则主叫终端直接向被叫终端发送建立连接请求(Setup)消息；

(4) 被叫终端直接向主叫终端回复呼叫开始进行(Call proceeding)的消息，表明收到了请求；

(5) 被叫终端为了加入通话，向网守发送 ARQ；

(6) 网守回复 ACF 或 ARJ；

(7) 如果得到允许，被叫终端直接向主叫终端发送 Alerting 消息，该消息等效于在 PSTN 上建立呼叫时听到的振铃信号；

(8) 最后，如果用户接受该呼叫，则被叫终端直接向主叫终端发送建立连接(Connect)的消息。

两个终端为了得到建立连接的许可而与网守之间交换的信息是**注册、认可、状态信令协议**(RAS)的一部分，两个终端之间直接交互的有关呼叫建立的消息是 Q.931 信令协议的一部分，而

RAS 和 Q.931 协议都属于 H.225 建议的一部分。

H.225.0 不仅描述了 H.323 终端之间，也描述了 H.323 终端与在同一 LAN 上的 H.323 网关之间的传输方法。在这里，LAN 可以是一个网段，也可以是用桥或路由器连接起来的企业网，但是使用过大的网（如几个互联的 LAN）会导致视听业务质量的明显下降。H.225.0 给用户提供方法以确定质量下降是不是由 LAN 拥塞而引起的，并提供采取相应对策的步骤。

从图 8-20 可以看到，音频、视频、应用数据和控制信号分别经 UDP 和 TCP 传输，因此，在 H.323 中这些媒体的复用和解复用操作由 UDP（TCP）/IP 层来完成；而音、视频通过各自的 RTP 会话传输，它们之间的同步则依赖于 RTP 时间戳来建立。

8.3.4 公用电话网视听业务标准（H.324）

H.324 标准应用于低码率的公共电话网上的视听业务终端。图 8-22 给出了 H.324 终端的协议参考模型。图中视频和音频信号分别采用低码率压缩算法 H.263＋和 G.723.1，它们分别是 H.263 和 G.723 的改进版本，具有更高的压缩效率。控制协议仍采用 H.245。为保证可靠性，传输控制消息的子通道具有差错恢复与重传机制，这可以通过使用 LAPM（V.42 Link Access Procedure for Modems）或 SRP（Simplified Retransmission Protocol）来实现。当建立起 H.245 控制通道后，可以进行能力集的交换，并为每种媒体建立任意数目的逻辑通道。对于像电子白板等这样共享应用的数据，可以通过 V.14 和 LAPM 等数据协议提供可靠传输。H.324 的呼叫部分遵从各国公用电话网的信令协议。各逻辑通道的复接遵循 H.223 协议。H.223 是一个面向连接的复接协议，它可将任意数目的逻辑通道复用到一条电路交换的信道上。

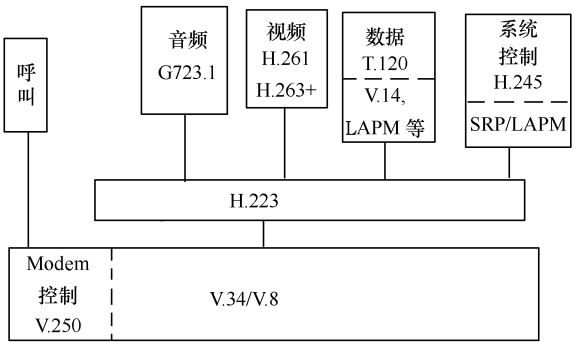


图 8-22 H.324 终端的协议参考模型

多路复接由多路复接层和适配层组成，适配层在复接层之上。适配层根据不同媒体的需要处理逻辑帧的形成、差错控制、包排序和重传等。在 H.223 中定义了三种适配层，其中 AL 1 用于变速率帧信息，它依赖于高层进行差错控制；AL 2 用于数字音频信号，AL 3 用于数字视频信号。三种适配层的特性不同，具有低延时的子信道允许较高的差错率；可靠的子信道则可能有较长的延时。为了减小延时，在 AL 3 中视频数据通常被打成小的变长包，例如 100 B 左右。每个包加 16 比特 CRC 以供检错。AL 3 还允许重传（可选项），但只有重传的信息能在它的解码时刻之前到达时，使用这个可选项才有意义。

在复接层中，经过各个逻辑子信道的适配层打包（逻辑帧）的音频、视频、应用数据和控制信号等按字节交替传送，形成一个流。图 8-23 表示 H.223 的复接原理，其中（a）给出一个复接字节流的结构。该流分为许多独立的段，每段称为一个**复接层 PDU**（MUX-PDU）。段间由 HDLC 标识 F（01111110）分隔。为防止数据流中出现的 01111110 被误认为标识 F，需要采用**比特填充**技术，即在数据流中填充非信息比特，例如在连续 5 个“1”之后填充 1 个“0”，以降低数据流中出现与标识 F

相同的情况的概率。在接收端，将填充比特去掉，恢复原数据流。此外，每个 MUX-PDU 开头有一个头字节 H，用来说明后面的信息字节 I 是如何排列的，即逻辑子信道的组成方式。图 8-23(a)表示出其中一个 MUX-PDU 内音频 A、视频 V、应用数据 D 和控制字节 C 的组合情况。值得注意，适配层输出的包并不一定能完整地放入一个 MUX-PDU，换句话说，适配层形成的一个逻辑帧可以拆分封装到几个 MUX-PDU 中去。

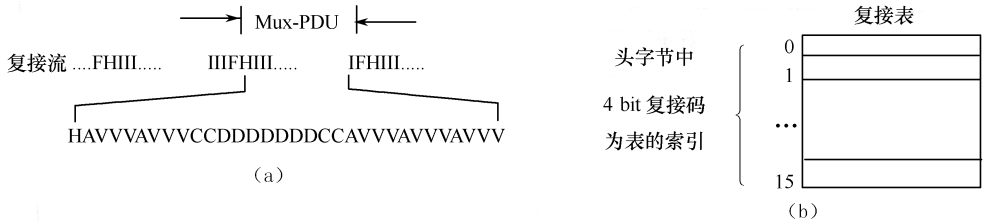


图 8-23　H.223 复接原理：(a)复接字节流结构；(b)复接表

H.223 的复接方式十分灵活，这体现在头字节 H 的使用上。H 中包含一个 4 比特的**复接码**。复接码是图 8-23(b)所示复接表的索引，发端和收端都拥有这个表(表由发端决定，在使用前通过 H.245 传送给收端)，表的每个条目表示逻辑信道的排列方式及每个信道占用的字节数。在通信过程中，发端通过简单地修改头字节中的复接码，能够很容易地快速改变对可用带宽的分配；收端根据接收到的复接码查表，即可将 MUX-PDU 中的字节正确地拆分成各个媒体流。

复接的比特流经 V.34 调制/解调器转换成模拟信号，以便在公用电话网上传送。当调制/解调器中的网络信令与数据传输的启动会话规程是分离的部件时，V.250 用于提供对调制/解调器(或网络)的控制。

8.3.5　移动电话网视听业务标准(H.324M 和 3G-324M)

1. H.324M

H.324M 是国际电联在 H.324 基础上针对移动电话网视听业务终端与系统提出的标准。H.324M 除用“无线接口”取代图 8-22 中的 V.34 调制解调器外，其余部分与图示的 H.324 协议参考模型相同。由于底层网络的变化，建立连接过程有一些变化，这些改变由 H.324 Annex C 定义。同时针对移动信道误码率高的特点，H.324M 做了如下两个方面的改进：

(1) 采用抗误码性能好的视频编码 H.263＋＋，它是 H.263＋的改进版本。我们将在 9.4 节对抗误码编码专门加以讨论；

(2) 提高 H.223 的抗误码性能，其成果总结在 H.223 Annex A/B/C/D 中，这是本节主要介绍的内容。

H.223 原来是以 V.34 连接为传输途径的。在 V.34 连接上，典型的误码率为 10^{-6}，这在相当长的时间(许多秒)内可以看做信道是无差错的，但对于无线信道，我们已经知道误码率可能高达 10^{-3}，因此有必要提高 H.223 传输的可靠性。改进后的 H.223 提供多个层次的复接层，使制造商可以根据不同的应用环境在抗误码的鲁棒性、实现的复杂性和代价等方面折中选择。Level 0 是原来的 H.223(见 8.3.4 节)，MUX-PDU 是可变长度的，并用 8 比特的标识 F 分开[见图 8-24(a)]。图中头字节中 MC 为复接码，CRC 为复接码校验，PM 为逻辑帧结束标记。标识符的长度太短以及为避免应用数据与标识符相同而采用的比特填充是导致 Level 0 对差错敏感的原因，因此在 Level 1 中使用了 16 比特长的同步标识符[见图 8-24(b)]，同时不再使用比特填充技术。虽然应用数据中也可能出现与 16 比特标识符相同的情况，但毕竟概率很小，不使用比特填充技术问题也不大。在 Level 2 中，除了也使用 16 比特的标识符外，在头信息中增加了复接载荷长度域 MPL[见图 8-24(c)]，这为接收端搜寻下一个标识符提供了进一步的信息。同时，整个头信息进行了 Golay

编码以便于检错。在 Level 1 和 2 中，MUX-PDU 的长度是整数个字节，而在 Level 0 中，由于比特填充，其 MUX-PDU 的长度有可能不是整数字节。在搜索同步标识符时，字节对齐的特征有利于减少遭遇信息比特与标识符相同的概率。

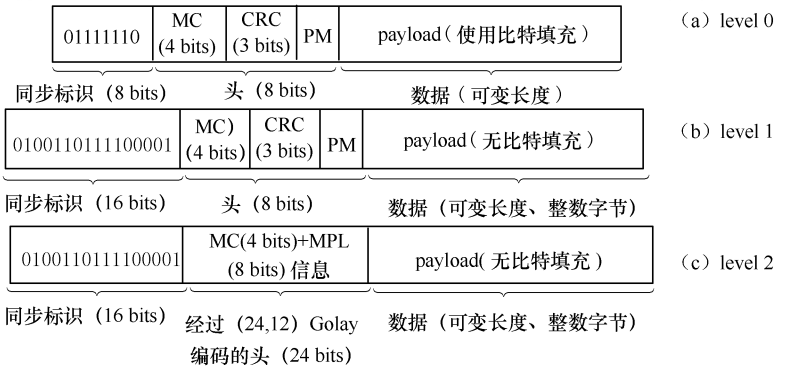


图 8-24　H. 223 复接层次

在 H. 223 的 Level 1 和 2 中没有对适配层作任何改变，而在 Level 3（Annex C）中，引入了针对高误码信道的 AL1M、AL2M 和 AL3M，分别用于应用数据和控制信号、音频以及视频。Level 3 的复接层与 Level 2 基本相同。AL1M 的特点在于它支持检错、前向纠错和重传。它的工作原理如图 8-25 所示，其中 AL-SDU 和 AL-PDU 分别表示适配层的输入和输出数据单元。

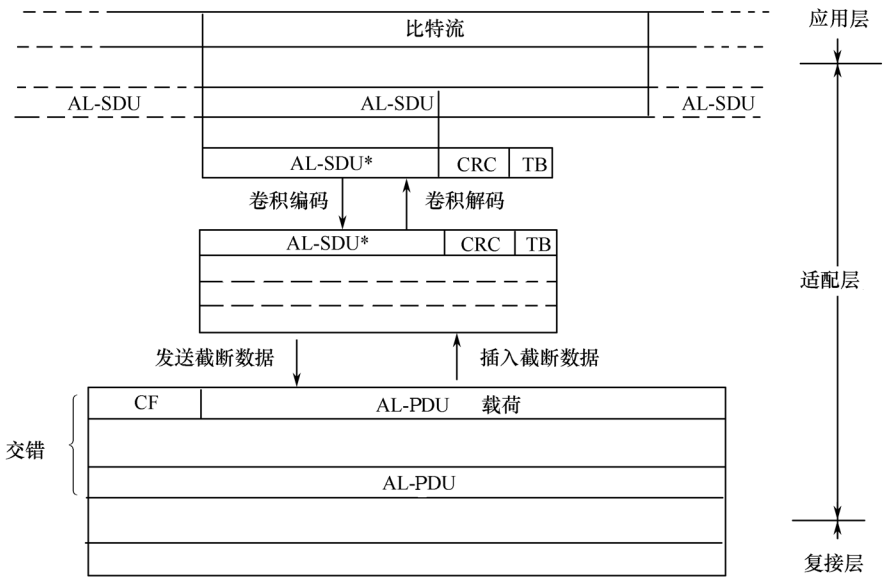


图 8-25　AL1M 工作原理

将 AL-SUD 的一部分（也可以是全部）加上 CRC 检验码和 4 比特的包尾（TB）作为信道编码器的输入，信道编码器采用码率兼容截短卷积码 **RCPC**（Rate Compatible Punctured Convolution code）。在进行 RCPC 编码之前加 CRC 的目的是，使接收端可以判断卷积解码后的数据是否还存在错误。加 4b 包尾是由于采用了码率为 1/4 的卷积码。卷积编码后的数据放在缓存器中，其中的一部分（截断）输出构成 AL-PDU 的载荷。控制域 CF 包括序列号、重传次数等，以及对它们进行 SEBCH 或 EGolay 编码生成的校验比特。最后，将 AL-PDU 载荷和 CF 一起作数据交错，以进一步提高抗突发错误的能力。关于数据交错我们将在 9. 3. 4 节中介绍。解码是上述步骤的逆过程。

此外，AL1M 还提供 2 种重传机制。

AL2M 只支持序列号(也采用 SEBCH 或 EGolay 编码)和 AL-PDU 的数据交错，两者都是可选项；而 AL3M 则和 AL1M 基本相同。

2. 3G-324M

3G-324M 是 3G PP 和 3G PP2 根据 H.324M 为 IMT-2000 制定的电路交换网络视听业务标准。3G-324M 终端协议参考模型如图 8-26 所示。它指定 H.263 为视频强制性标准，MPEG4(而不是 H.264)simple @ level 0 为推荐标准。在语音方面，指定 AMR(Adaptive multi-rate)编码为强制性标准，G.723.1 为推荐标准，提供 4.75～12.2 kb/s 的传输速率。在控制信道上，使用了控制信道分段和重组层 CCSRL(Control Channel Segmentation and Ressembly Layer)以及编号简单重传协议 NSRP(Numbered Simple Retransmission Protocol)，以保证 H.245 消息的可靠传输。在复接层上，采用 H.223 Annex A 和 B(Level 1 和 2)，而 Annex C 和 D(level 3)是可选项。有关编码和呼叫控制的详细规定可参考 3G TS 26.111 和 TS 26.112 规范。

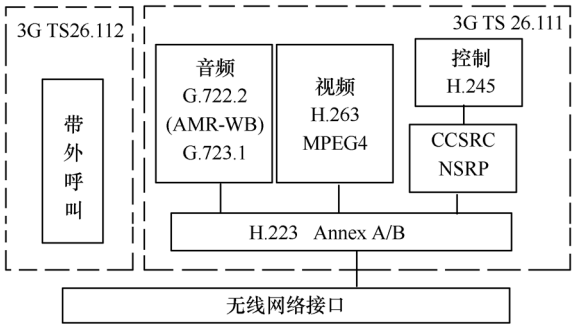


图 8-26　3G-324M 终端协议参考模型

8.3.6　视听系统的通信控制协议

通信控制的主要功能包括：能力交换与通信模式的确定、子通道(逻辑通道)管理、身份认证、密钥分发、动态模式转换(参加、退出会议、速率转换等)、远程应用功能控制、流量控制和多点控制/响应等。所谓能力交换，是指通信的双方将自己的能力(如总传输速率、是否具备语音和视频信号同时通信的能力、可处理的压缩编码/解码方式、数据传输是否采用 T.120 协议、实时媒体流采用的是 PS 还是 TS，以及 AAL 类型等)互相交换、协商，以确定此次通信采用哪些模式。通信过程中 QoS 的缩放(如传输速率变化时相应地改变分辨率或帧率)通过动态模式转换实现。远程应用功能包括对远端摄像机的控制、凝固图像、快速刷新、静噪以及维修时信号的环回等。而多点会议控制则包括会议的申请、加入和退出、发言权控制、主席控制、数据令牌控制和同时打开多个会议等。

ITU-T 最早制定的控制协议是用于 H.320 系统的 H.242/H.243/H.230(见表 8-4)，而用于其他 H 系列系统的控制协议为 H.245。此外，针对多媒体会议中的应用数据传输，即对共享数据的控制，ITU-T 还制定了 T.120 系列的协议和近年出现的 H.239 协议。下面对这几种控制协议分别做简单的介绍。

1. H.242/H.243/H.230

系统控制协议 H.242 主要涉及对每个流的带宽协商。终端与 MCU 建立连接之后，终端内的系统控制就向 MCU 的多点控制部分通知自己的能力，如视/音频格式和压缩标准，以及每个信道所建议的比特率等。MCU 从这些能力中商定一个最小集，以使各成员都能参加会议。

在 8.2.1 节对 H.221 的分析中我们已经了解到，公务子信道中的 BAS 码携带着通信控制的

消息，而 H.242/H.243/H.230 协议则规定了实现控制的过程，二者必须结合使用。由于一个 BAS 码只有 8 比特，所能表达的消息数目有限，要传送较为复杂的控制信息时，需要采用单字节扩展(SBE)或多字节扩展(MBE)的 BAS 码。不过，尽管可以扩展，使用 BAS 码的方式所能表达的控制消息数目仍然是有限的，此外扩展也不够灵活。

由于 BAS 码是与连续媒体复接在一个帧内传输的，连续媒体对误码的要求不高，但控制信息的传输却需要保证可靠性。因此一个 BAS 码除了 8 个信息比特外，还加上了 8 个纠错比特，以使其在有一定误码的条件下能够正常工作。

H.242/H.243/H.230 能够实现的主要控制功能除能力交换外，还包括通信模式确定、模式转换、远程应用功能控制和多点会议控制。早期的视听业务着重于语声和会话者图像的传递，数据的交互是很少的。因此这组协议具有良好的实时性，发送端和接收端可以同步地进行模式转换，适合于对连续媒体流的通信控制，但是它对应用数据的多点控制能力较差。这组协议广泛地应用于会议室型的系统，实现起来也比较简单。

此外，这组协议还定义了加密控制子信道(ECS)，采用 H.233 和 H.234 进行加密。

2. **H.245**

H.245 是 H.310、H.323、H.324 和 H.324M 系统的控制协议，同时它也在 V.70(使用调制/解调器的)多媒体终端中采用。它的设计思想与 H.242 有显著的区别。首先，它与复接标准不相关，通信控制消息和通信控制过程均在 H.245 中定义。其次，控制消息在一个专有的逻辑通道中传送，该通道在通信一开始就打开，在整个通信过程中不关闭。采用专有的逻辑通道显然可以比 BAS 码传送更多种类和更复杂的控制信息。同时，该信道总是建立在可靠的传输层服务之上(见图 8-20、图 8-22 和图 8-26)，因此在定义控制消息和过程时不需要考虑差错控制。

H.245 分为 3 个基本部分，即句法、语义和过程。第一部分是用 ASN.1 定义的控制消息句法。语义部分描述句法元素的含义，并提供句法的制约条件。过程部分则用 SDL(Specification and Description Language)图定义交换控制消息的协议。在句法中规定了扩展标志和标识版本号的协议标识(Protocol Identifier)域，以便于将来对 H.245 功能进行扩展和应用于新类型的系统。过程部分不仅定义了正常操作，还定义了异常事件的处理。

H.245 实现的控制功能主要有：能力交换与通信模式确定、对特定的音频和视频模式的请求及模式转换、逻辑通道管理、对各个逻辑通道比特率的控制、远端应用控制、确定主、从终端和修改复接表等。不同的控制功能对应于不同的实体，每个实体负责产生、发送、接收和解释与该功能有关的消息。实体间相对独立，相互之间只通过与 H.245 的使用者间的通信进行联系(见图 8-27)。模块化的结构与封装使得 H.245 具有良好的扩展性和实时性。

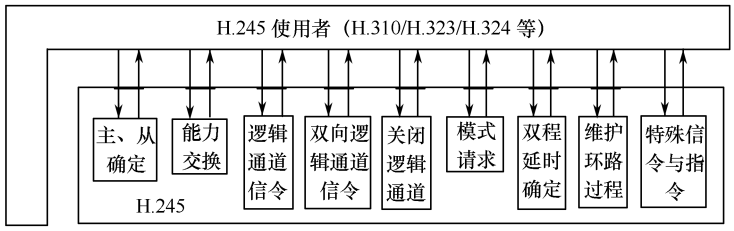


图 8-27　在 H 系列终端中的 H.245 实体

H.245 有管理多个逻辑信道的功能。逻辑信道可以是单向的，也可以是双向的。每一个逻辑信道号代表一个特定的信道。在进行能力交换之后、多媒体数据的实际传输之前，终端通过逻辑信道信令(打开/关闭)为编码/解码分配资源。要求打开一个逻辑信道的请求中包含着对所要传输的数据类型的描述(如 6 Mb/s 的 H.262 MP@ML)，此信息提供给接收端以分配解码资源。发送端

收到接收端的肯定确认信号之后，才正式开始数据的传送。接收端也可以拒绝发端建立逻辑信道的请求。多个逻辑信道可以按复接标准复接成单一的比特流，打开逻辑信道的请求消息中标明该信道复接时的位置，例如在传送 MPEG2 时，标明传送流的 PID 或节目流的 Stream-ID，以及标明携带 H.222.0 复接传送流的虚通道等。可以看出，逻辑信道的建立与在该信道上传送的数据类型是相关联的，因此 H.245 既适合于对连续媒体的控制，也适于对突发数据和大块数据流的通信控制。

我们在 8.3.4 节中已经了解到复接表的概念，描述信息流复接方式的复接表是通过 H.245 传送的。假设一个复接表包含 16 个复接项，即 16 种复接方式，转换复接模式（如网络拥塞需要转换到低速率模式）时，复接层只需要在这 16 种方式中选一种即可，因此模式转换速度很快。H.245 考虑了加密控制，可以打开/关闭加密控制逻辑信道。在模式转换的速度和加密控制两个方面，H.245 与 H.242 同样具有较好的性能。

3. T.120 系列

T.120 系列标准是 ITU-T 针对音图（Audiographic）会议而制定的多点通信控制协议，它由一组协议构成。我们在对 H 系列系统的分析中看到，它通常作为可选项对应用数据（Data/Telematic）进行控制。

T.120 协议的系统模型如图 8-28 所示，其中由 T.123、T.122/T.125 和 T.124 构成的**通信基础设施**（Communication Infrastructure）是整个体系结构的核心。

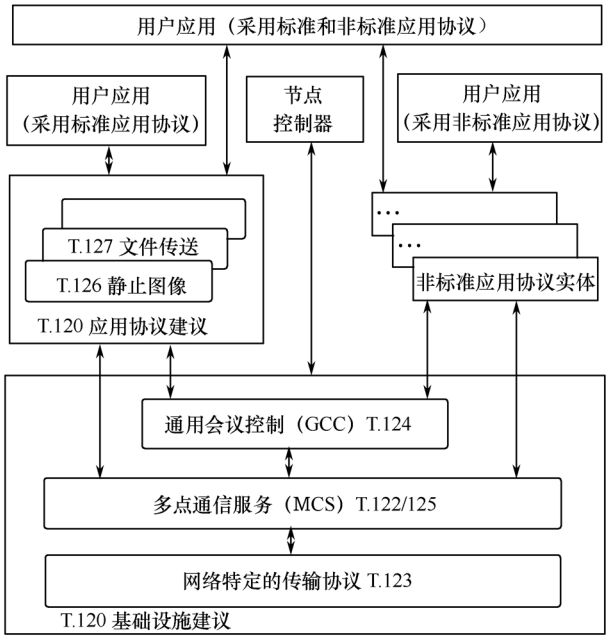


图 8-28　T.120 协议系统模型

T.123 负责向上层提供可靠、有序和具有流量控制的数据传输服务，对数据单元没有长度的限制。T.123 规定了不同类型网络的协议栈配置，使会议可以建立在 PSTN、同步电路交换和分组交换等多种网络之上。

T.122/T.125 定义了**多点通信服务** MCS（Multipoint Communication Service）。MCS 支持任意多个应用实体之间的全双工多点通信，这些互联的应用实体可以存在于不同的网络环境之中。MCS 将若干个点到点的传输层连接组成**多点域**（Multipoint Domain），每个多点域对应于一个独立的会议，而

域内的数据传输是通过 MCS 通道来完成的。MCS 通道可以分为静态和动态两种类型。静态通道随着多点域的创建而产生，而动态通道是在会议进行过程中，根据应用实体的申请而创建的。动态的 MCS 通道又可以进一步划分为多点传输通道、单一成员通道和私有通道。应用实体可以利用 MCS 的服务加入或脱离一个通道。MCS 按“树”的方式将多点域内的所有节点组成一定的层次结构，顶节点利用令牌完成对各种资源的分配与管理。

T. 124 提供**通用会议控制** GCC(Generic Conference Control)服务，具有创建、控制和终止会议的功能。T. 120 协议还定义了一些具体应用的标准，如静态图像传输(T. 126)、多点的二进制文件传输(T. 127)等，并在 T. 121 中提供了一般性应用模板(Generic Application Template)，其目的在于使具体应用的开发者能够更充分地利用 MCS 服务和确保不同应用之间的互操作性。

可以看出，T. 120 多点通信的控制能力是很强的。通过 T. 122/T. 125 的 MCS 域管理可以灵活地组网；通过分层的 MCS 结构和顶节点可以方便地组成多级多点会议；通过 MCS 域融合和域分离可以将两个同时召开的会议合成一个，或将一个会议分成两个。MCS 还支持多个同时召开的会议。但是，由于协议的复杂性，T. 120 不宜应用在实时性要求较高的连续媒体传输中，这就是在 H 系列系统中只采用它作为应用数据通信控制的原因。

为了弥补 T. 120 的不足，ITU-T 曾制定过与 T. 120 配合使用的、对实时媒体进行通信控制的 T. 130 系列标准。由于 T. 130 应用不广泛，我们在这里不作介绍。T. 120 系列更多的是作为数据控制协议在 H 系列系统中使用。

4. **H. 239**

在多媒体会议中，除了与会者相互看得见、听得着之外，常常需要共同观看一些资料，如幻灯片、电子白板、或计算机屏幕上的文档、图表等。这些画面虽然可以作为用户数据或静止图像通过 T. 120 系列建立的数据会议实现共享，但是由于 T. 120 系列协议的复杂性，ITU-T 于 2003 年制定了 H. 239 协议，以较为简单的技术来支持上述应用。和本节前面讨论的几个协议不同，H. 239 不能称为真正意义上的通信控制协议，它可以看作是 H. 245 和 H. 320 系统的控制协议的扩展。

在会议期间除了传送与会者画面的视频通道外，再建立一个专门的视频通道传送幻灯片、白板等图像是实现资料共享的一个简单手段。对于使用 H. 245 的 H. 300 系统，通过 H. 245 可以建立多个视频逻辑通道；而在 H. 320 系统中只能建立一个视频通道，因此 H. 239 的第一个功能就是提供一种方式使 H. 320 系统能增加一个(或几个——在制定中)视频通道。H. 239 的第二个功能是通过**“角色标记”**为视频通道指定所扮演的角色。“角色”定义了该通道中视频流的用途和终端(或 MCU)对它的处理方法，例如传送与会者画面的通道的角色为“实况(live)”；传送共享资料的通道为“演示”(presentation)。通常“实况”视频通道是双向的；“演示”视频则是单向传输的，以便降低终端处理和 MCU 分发的复杂性。因为每个与会者都有可能要做演示，所以“演示”角色是由令牌控制的，H. 239 规定了令牌的传递和管理方法。逻辑通道角色标记的指定在收发双方进行能力集协商时完成。

8.3.7 H 系列系统间的互通

工作在不同网络上的终端之间应该能够互通，这是 ITU-T 制定标准中考虑的重要目标之一。图 8-29 是一个 H 系列系统互通环境的示意图。在这个环境中，H. 323 是互通的基准，不同的 H 系列终端通过不同的网络连接到 H. 323 的域中。

网关(Gateway)在不同系统的互联中扮演着极其重要的角色。首先它通过对不同信令信号的转换在不同类型终端之间建立(或撤销)通信链路；其次在通信过程中，它负责不同协议下的控制命令、复接方式和媒体编码格式的转换。ITU-T 为这些转换制定了 H. 246 标准。图 8-30(a)、(b)和(c)分别给出了几种基本的 H. 300 系统之间信令、控制区和复接协议转换的对应关系。互通的终端应

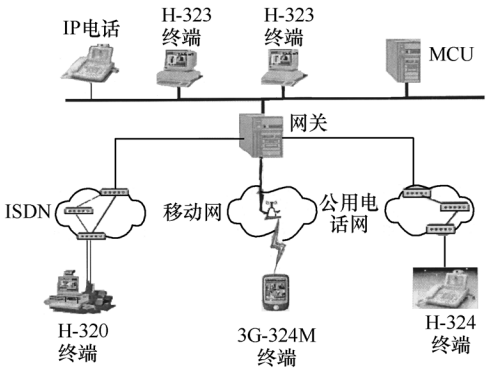


图 8-29　H 系列系统间的互通

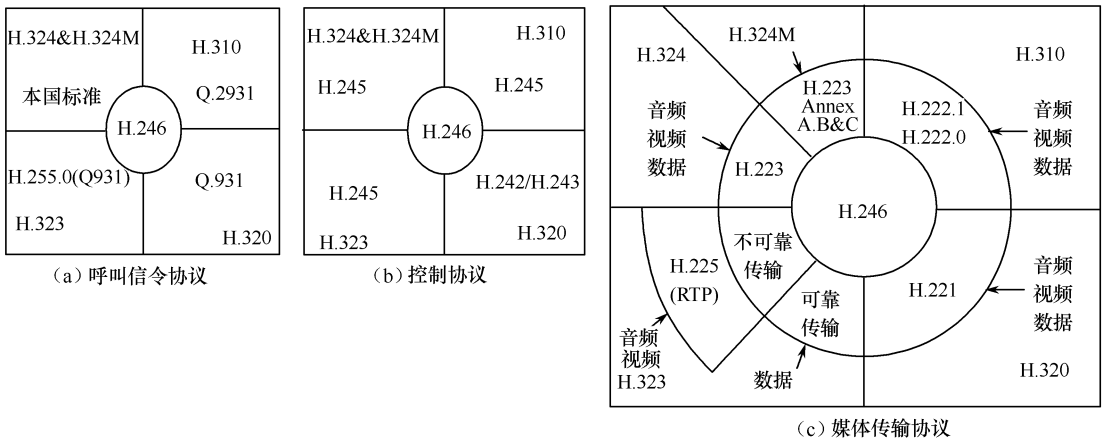


图 8-30　H 系列系统间的互通

当采用相同的音频、视频和数据格式，以避免在网关中进行编码方式的转换而引起音、视频信号质量的下降和附加延时。

8.4　会话发起协议和会话描述协议

8.4.1　会话发起协议(SIP)

会话发起协议 SIP(Session Initiation Protocol)最初是 IETF 针对在 IP 网上建立多媒体会话业务而制定的一组协议中的一个，这组协议包括：会话发起协议 SIP(RFC 3261)、会话描述协议 SDP(RFC 2327)、会话发布协议 SAP(RFC 2974)和简单会议控制协议 SCCP(Simple Conference Control Protocol，draft-IETF)。SIP 经过扩展后可以在更多的多媒体应用中使用，如即时通信、网络游戏、邮件或其他事件的通知等。

由于 SIP 的简单、灵活和扩展性好的优点，它已为多个国际组织，如国际电联、3GPP/3GPP2 等所接受，如 7.11.2 节所述，经过扩展的 SIP 已经成为 IMS 中的**信令协议**。在本节中，我们虽然将从与 H.323 比较的角度引入 SIP，但是必须认识到它实际上是下一代网络中最重要的通信协议之一，因此我们单辟一节来对它加以讨论。

在 8.3.3 节中我们看到 H.323 是国际电联提出的分组网视听业务标准，它从电信的角度出发，将一些电信协议修改之后应用到 IP 网络上，例如它完全引用了 N-ISDN 的 Q.931 呼叫信令协议。但是在 IP 网络上建立多媒体会话的机制与电信网的情况并不完全相同。在电信网中，呼叫建立的过程也

就是在 TDM 网络中预留话路(时隙)的过程,一旦呼叫连接成功,就可以开始端到端的通信。而对于建立 IP 多媒体会话来说,还需要获得双方的 RTP 地址、进行能力集(如编码方式等)协商等,Q.931 并不具备这些功能,因此在 H.323 中,呼叫成功之后只是打开了一个 H.245 的控制子信道,通过这个子信道进行 H.245 的消息交换才能完成上述功能,从而打开音、视频子信道开始端到端的通信。这就造成了 H.323 会话的建立过程十分复杂。

SIP 可以替代图 8-20 中 H.323 的 RAS、呼叫建立(Q.931)和系统控制(H.245)协议。而且 SIP 信令消息在 IP 网中的传输路径和多媒体数据流的传输路径是相互独立的,这给 SIP 的应用提供了很大的灵活性,使它能够应用于音视频、数据等多种会话以及网关之间的通信。从第 7 章我们看到,无线网络和有线网络都在向全 IP 化演进,这为以 SIP 为信协议的视听业务(如可视电话与会议)展现了十分广阔的前景。

SIP 是一个应用层控制协议,处于传输层之上。SIP 消息可通过 TCP、SCTP 或 UDP 传输。SIP 支持多播,可以使用多播支持会议呼叫。

1. SIP 地址

SIP 使用统一定位标识符 URI(Uniform Resource Identifier)来唯一地标识终端,SIP URI 由用户名和主机名两部分组成(用户名@主机名)。为了便于使用,与电子邮件的地址类似,SIP 地址由用户名和域名表示,如 SIP:Zhang@bupt.edu.cn,表示 zhang 在北京邮电大学的 SIP 地址。SIP 还提供安全 URI,例如 SIPS:Zhang@bupt.edu.cn。使用 SIPS URI 时,从主叫端到被叫端的所有 SIP 消息都经过**安全加密传输层 TLS**(Transport Layer Security)传输。经扩展后,SIP 还可以支持其他类型(如普通电话号码等)的地址。每个终端须向本域内的注册服务器注册。注册时,注册服务器完成 SIP 地址和 IP 网中标识主机地址和应用程序的 URI 之间的绑定,同时将此信息记录在**位置服务**的数据库中。

2. SIP 会话建立

与每个电话局中有一台本地交换机一样,每一个域中有一台 **SIP 代理服务器**,负责完成 SIP 消息的路由。图 8-31 给出 SIP 终端通过代理服务器进行呼叫的过程。主叫终端中的**用户代理客户 UAC**(User Agent Client)首先向本地代理服务器发出 Invite 请求消息,请求消息的头字段中包含本次会话的标识符、主叫地址、被叫地址和希望建立的会话类型等。图中主叫和被叫地址分别为 SIP:Zhang@bupt.edu.cn 和 Wang@macro.com。代理服务器 A 收到请求后,向主叫方回复一个响应消息(Trying),说明收到并正在处理 Invite 消息。代理服务器 A 通过位置服务查询被叫方是否在本域内,如果是,将 Invite 直接转发给被叫方;如果不是,则通过特定类型的**域名服务 DNS**(见 RFC 3263)或**轻量级目录接入协议 LDAP** 等提供的位置服务来获得被叫终端所在域的代理服务器 B 的 IP 地址,将 Invite 转发给 B,并在转发前将自己的地址加进 Invite 头的 Via(经由)字段中。B 收到请求消息后,向 A 返回响应消息 Trying,并通过位置服务找到被叫方的 IP 地址(及端口号和传输协议),在 Invite 头的 Via 字段中加入自己的地址后,将请求消息转发给被叫终端的**用户代理服务器 UAS**(User Agent Server)。

被叫端收到 Invite 消息后发出振铃,并向 B 返回 Ringing 的响应消息,此消息又经 A 后到达主叫端。由于被叫端从 Invite 消息的 Via 字段知道了途径服务器的地址,所以 Ringing 消息无须通过 DNS 和位置服务就能返回到主叫端。在回传过程中,每个代理服务器根据 Via 字段得知下一跳的地址,并删去自己的地址。当 Wang 拿起话机时,被叫终端(UAS)返回 OK 响应消息;如果被叫终端忙或无人应答,则返回一个错误消息。假设 OK 消息返回到主叫端(UAC),主叫端送一个 ACK 消息给被叫端确认连接的建立。由于主叫端和被叫端已经通过 Invite/OK 消息交换的头信息知道相互的 IP 地址,因此从 ACK 之后消息交换不再经过代理服务器 A 和 B。至此,通过

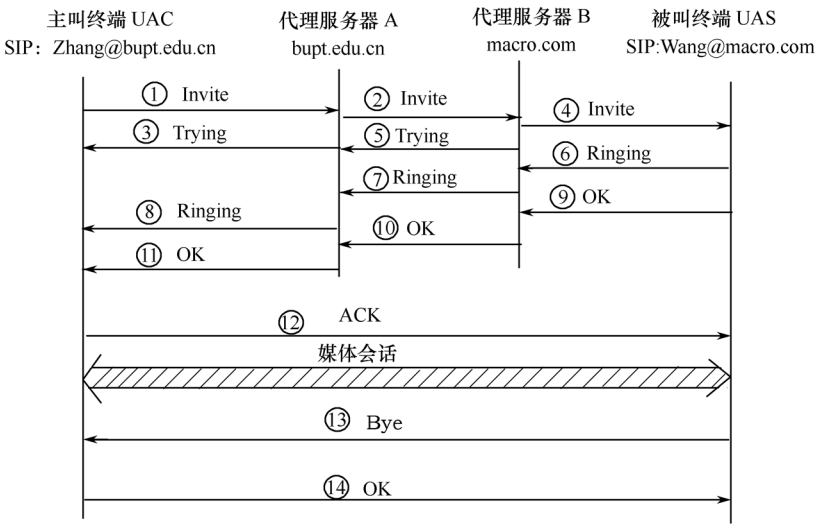


图 8-31　SIP 会话的建立和拆除

Invite/OK/ACK三次握手 SIP 会话建立。

会话建立后，主叫和被叫双方开始进行媒体数据的交换。媒体数据交换直接采用通话双方的 IP 地址，而不必使用原来 SIP 信令消息的路径(如图 7-44 曾给出例子)。会话结束时，Wang 挂机，被叫端送出 Bye 消息；主叫端则以 OK 回应，终结此次会话。

值得注意：(1)一个 SIP 用户可以注册多个 SIP 网络终端(通常在一个域中)。在这种情况下，代理服务器可按优先级逐个尝试建立会话，如在上例中，如果 Wang 的电话忙，代理服务器 B 可以将从 Zhang 来的 Invite 消息转发给 Wang 的视频电子邮件服务器，这称为**呼叫转移**。另一种选择是 B 将 Invite 消息同时转发给这两种终端，此种操作称为**分支**(Forking)。(2)SIP 是一个客户端/服务器协议，如前所述，发起方称为客户(UAC)，响应方称为服务器(UAS)。在会话建立过程中，代理服务器需要对终端发送来的请求做临时响应；而相对下一跳代理服务器，代理服务器本身是一个 UAC。因此代理服务器同时具有 UAC 和 UAS 功能。

3. 重定位功能

图 8-32 表示一个 SIP 重定位功能的例子。由于位置的可移动性，用户可能注册多个不在同一域内的 SIP 地址，例如图中 Wang 有 Wang@macro.com 和 Wang@micro.edu 两个地址。当主叫方 Zhang 呼叫 Wang 时，macro 域中的 SIP 代理服务器 B 发现 Wang 不在本域中，则向**重定向服务器**发出请求(见图中③)。重定向服务器通过位置服务得到 Wang 的另一 SIP 地址，并将此地址告知主叫方所在域的代理服务器 A(④)。A 通过 DNS 获知新地址域代理服务器 C 的地址，并将主叫端的 Invite 消息直接转发给 C(⑤)。C 再通过本域的位置服务获得 SIP：Wang@micro.edu 的 IP 地址，将 Invite 消息转发给被叫端(⑥)。在图 8-32 中为了清楚起见，我们没有标明相关的响应消息。

当用户注册有多个位置的地址时，重定向服务器返回一个位置列表，由代理服务器进行所有的用户定位的尝试。

4. 会话模式更改

SIP 允许在通话过程中对会话的特性进行修改，例如当网络条件变坏时，通话的一方希望关闭视频、只使用音频继续通话，这可以如下方式实现：希望更改会话模式的一方向对方送 Invite 消息，该消息包含对新模式的描述，并且标明当前会话的标识，使对方知道是一条更改消息，而不是一个

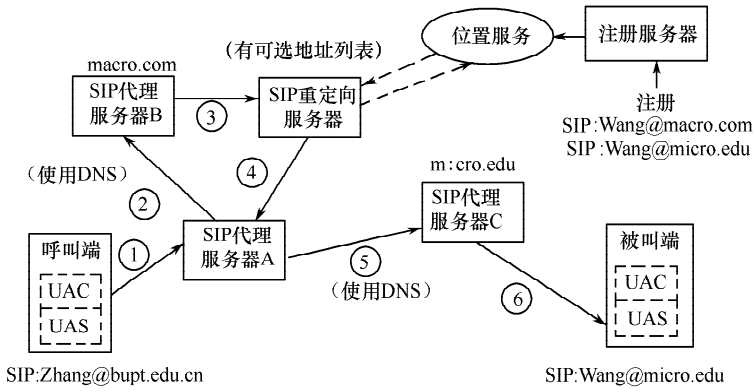


图 8-32　重定位呼叫过程

新的 Invite 请求。收到 Invite 消息的一方回复 OK 消息表示接受更改，回复出错消息表示拒绝。无论对哪种回复，发出更改请求的一方都响应 ACK。若对方接收请求，会话以新模式进行；若拒绝，当前会话仍以原模式进行，并不会中止。

8.4.2　SIP 消息格式和会话描述协议(SDP)

我们知道在分组网上建立多媒体会话时，除了建立传输层的连接外，很重要的还必须获得双方的 RTP 地址和进行能力集的协商。在 SIP 中这是通过 SIP 消息的**消息体**来完成的。

图 8-33 给出 SIP 消息的格式，它主要分为起始行、消息头和消息体 3 个部分，其中消息头和消息体之间用空行隔开。对于请求消息，起始行给出请求消息的类型(如图中 Invite)；对于响应消息，起始行给出请求是否成功及失败类型、原因。消息头所提供的信息便于代理服务器对消息进行处理。图的左侧给出 Invite 消息的一部分头信息，其中 Call-ID 为会话的标识符，From 和 To 与电子邮件中类似，给出主叫和被叫方的 SIP 地址，Via 字段包括从主叫到沿途各代理服务器的 SIP 版本号/传输层协议、IP 地址和端口号等，从而为响应消息提供返回的路径。在 8.4.1 节中我们已讲到请求消息每经过一个代理服务器，该服务器就把自己的地址信息加进 Via，而响应消息在返回过程中每经过一个代理服务器，该服务器就将自己的地址从 Via 中删除。Contact 字段给出呼叫转移或重定位操作所使用的同一用户的多个终端的 IP 地址。Content-Type 和 Content-Length 分别表示消息体所包含的内容类型(即会话的类型)和消息体的长度。

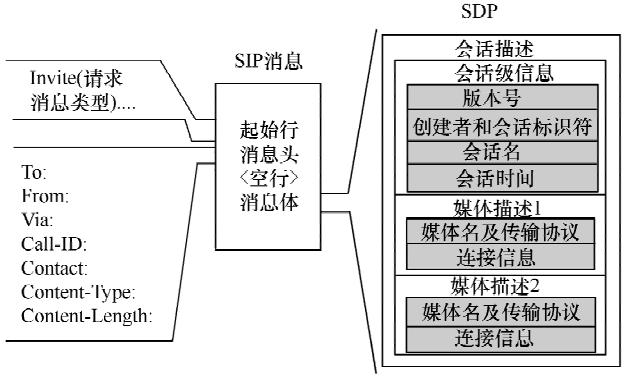


图 8-33　SIP 消息格式

SIP 会话中编码格式的协商和 RTP 地址交换等通过消息体进行。消息体是采用**会话描述协议 SDP**(Session Description Protocol)对会话进行描述的一段文本。SDP 是一种文本描述语言，由

IETF RFC 2327 所规定。SIP 消息携带 SDP 消息体就像电子邮件携带附件和 HTTP 消息携带 Web 页面一样。图 8-33 的右侧给出 SDP 消息体的结构，它包括会话级信息和媒体描述两部分，图中列出了每一部分的主要字段。会话级参数包括会话名、会话创建者及会话活跃时间等，而媒体参数包括信息流类型（如音频或视频）、端口号、传输协议（如 RTP）、编码格式等。

主叫方通过 Invite 的消息体说明它欲发起的会话的详细信息（如编码格式等）；被叫方通过 OK 的消息体回复它愿意接受的会话的类型。这样两个步骤就完成了双方对能力集的协商。如果被叫方不支持主叫方提出的媒体格式，主叫方在收到这样的响应消息后，必须重新生成新的 Invite 消息。

8.4.3 SIP 与 H.323 信令的比较

使用 SIP 作为信令协议的视听终端与 H.323 终端相比，除信令（RAS、Q.931 和 H.245）外其余部分（音、视频及应用数据传输）均可不变。表 8-5 给出 SIP 和 H.323 信令的一个简单比较。

表 8-5 H.323 信令与 SIP 的比较

	H.323 信令	SIP
设计思路	从传统电话模式移植到 IP 网上	将多媒体会话当作因特网的一个应用，为其增加信令
用户定位功能	无	有
信令多播	无	有
消息编码	ASN.1 二进编码需使用特殊的代码生成器进行语法分析	基于文本，简单易懂
会话协商	通过 Q.931 建立控制信道，再通过 H.245 协议协商，建立时间较长	在呼叫建连过程中利用 SDP 描述的消息体进行协商
会话管理	由 MCU 集中管理时便于带宽管理，但对大型会议易形成瓶颈	分布式管理，不易计费
传输层协议	信令信道/TCP	UDP，TCP，SCTP

值得指出，在 H.323 协议的最新版本中，会话建立和协商的时间已经缩短。但是 SIP 由于它的良好特性，如信令消息与媒体传输相互独立，并与底层网络的具体形式无关（只要支持 IP）；能够重用因特网的许多协议（如 DNS、RTP 等），无须建设新的基础设施；适用于任何 IP 设备（如终端、网关等）；扩展性好等，使其成为不仅是视听系统，而且是下一代通信网的信令协议的强有力的竞争者。

8.5 协同计算与组通信

8.5.1 协同计算的概念

协同计算（Collaborative Computing），或称为**计算机支持的协同工作** CSCW（Computer Supported Cooperative Work），是指两个或两个以上的人使用各自的计算机合作完成一项工作、或共同解决同一个问题。能够支持这样的组工作的系统被称为**群件**（Groupware）。群件的实质在于为协同工作的合作者创造一个共享的空间，以使他们能有效地完成同一项工作。

CSCW 通常通过对某些特征的区分来进行分类。图 8-34 给出了从时间、空间和用户规模进行分类的情况。从时间上来看，CSCW 分为异步和同步的两种。在异步 CSCW 中，合作者的协作活动发生在不同的时间段内，例如电子邮件发信人的发信时间和收信人的读信时间是不同的；而同

步 CSCW 中的协作活动则发生在同一时间内，例如在电视会议、即时通信或实时决策系统中，合作者实时地进行交谈和讨论。从地点上来看，协作可能发生在同一地点，例如在同一房间中进行的会面；协作也可能在处于不同地域的合作者之间进行，这些合作者通过 LAN、或 WAN 进行通信。从用户规模上看，有一个用户与另一个用户之间的协作，也有两个以上的用户、或多个用户组之间的协作。除了以上三种分类方法外，CSCW 的分类还可以依照控制方式（集中、或分散）和合作意识的强弱等进行，这里不再一一介绍。

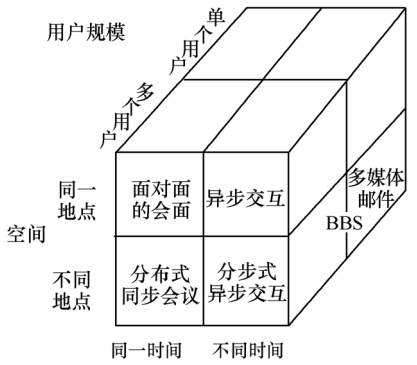


图 8-34　协同计算的分类

人们通常希望群件能够从非协作型的单用户模式无缝地过渡到两个用户之间和多个用户之间的协作模式，例如希望多用户协同工作的文字编辑器与单用户独立使用的编辑器具有类似的界面和工具。但是用户规模的任意扩展并不是一个容易达到的要求。

对 CSCW 的研究始于 20 世纪 80 年代中期，在随后的年代里，CSCW 逐渐形成了一个与分布式系统、计算机通信、人机交互、人工智能和社会科学等多门学科相关的研究领域。近年来随着多媒体技术和高速网络技术的发展，CSCW 与电视会议结合产生了功能更强大的分布式同步协同计算系统，这就是我们在第 1 章中讨论的多媒体合作工作系统。在这样的系统中，由于合作者之间听得见、看得着，增强了人和人之间的交流。同时，在这类系统的共享工作空间（如白板、共享的编辑器等）中，操作是实时的，操作的对象不仅包括静态媒体，也可以包括实时媒体。本节讨论的主要是这类系统。

8.5.2　组通信

组通信（Group Communication）是指能够支持同步（或异步）协同工作的多个用户之间的通信。图 8-35 给出了一个组通信的支撑模型。由图看到，组通信是建立在支持多播的网络之上的。多播提供了有效地完成一点至多点、或多点至多点数据传输的手段，而构成协同工作的环境还需要组通信来支持。图中的组通信代理由**组聚集**（Group Rendezvous）、会议和应用共享三个主要部分构成。

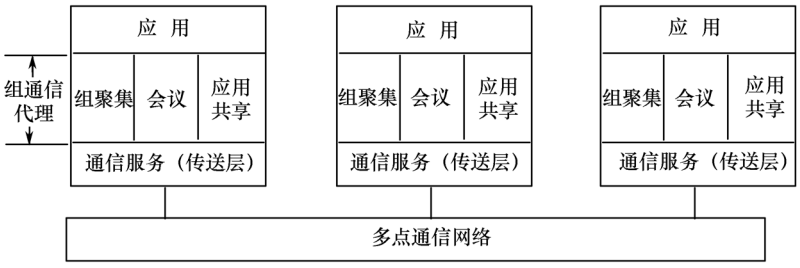


图 8-35　组通信支撑模型

组聚集向用户提供组织组活动的方法，以及获取关于组成员、正在进行的组活动和其他静态或动态信息的方法。组聚集通常分成同步和异步的两类。同步方法使用目录服务和显式邀请（Explicit Invitation）。通过目录服务（如 X.500）用户可以获取存储在信息库中的有关组活动的名称、参加者、参加者的角色和权限等信息。所谓显式邀请是指组活动的发起者需要通过点对点、或点对多点的方式向合作者发出邀请，才能组织起组活动。异步组聚集可以通过电子邮件或公告牌（Bulletin Boards）来实现，即将正在（或将要）进行的组活动及其参加者的有关信息放在邮件中或公布在公告牌上。此外，WWW 也提供了一种查询组活动信息的方法。

组通信代理的会议部分对实时媒体的通信进行管理。组活动参与者的图像和声音的交换是十分重要的，因为它创造了参考者“面对面”会见的感觉。对于视频和音频数据的传输及通信控制已在8.3节中做了详细的讨论，在这里我们仅就会议（组活动）的控制方式做一些补充。会议控制方式一般分为集中式和分布式两种：

(1)在集中式会议控制中，发起者首先向合作者发出显式邀请（这暗含着发起者已知合作者的地址，而这些地址可以从中央目录服务器中获得），每一个合作者要回复发起者它是否愿意参加会议。然后在愿意参加的合作者之间进行会议有关策略的协商，同时将有关会议的信息存入中央信息库，并利用可靠的消息传递机制将共享的会议状态信息通知所有参加者。这种控制方式的优点是控制方便，而且对每一个参与者而言，会议状态信息是一致的。其缺点是，当有一个新成员加入会议时，必须向每一个成员发送会议状态变更的消息，在新的会议状态建立之前有较长的延时。此外，在会议期间如果某一个参与者的链路发生故障，会议状态将被破坏，即使该链路恢复，会议状态也不见得马上恢复。因此，集中控制方式比较适合于规模较小、组成员不变动的静态组活动。

(2)在分布式会议控制中，会议的发起者首先通过多播实体建立一个多播空间向合作者发布有关会议的消息，会议就算建立了。愿意参加会议的合作者只需加入到特定的多播实体（如多播地址）中去，并通过组聚集发布它已参加会议的消息即可。在分布式控制方式中，会议状态是由每一个参与者通过多播将各自的状态传送给其他的参与者而建立的，由于在此种方式中对组成员没有全局的管理，因此每一个成员所了解到的会议状态可能是不同的。为了保持一定的一致性，会议状态消息要利用非可靠的消息传送机制周期性地传送。这种松散的分布式控制方式的优点是用户规模容易扩展，同时对故障的容忍性好，在某一段通信链路的故障排除之后，很容易重新建立起新的会议状态；其缺点是会议的参与者对会议状态的了解可能是不同的。松散控制比较适合于在WAN范围内、组成员经常动态变化的情况。MBone中的简单会话（Light-Weight Sessions）就是分布式控制的一个例子。

组通信代理的第三部分是对共享应用的控制。如前所述，群件的实质在于为协同工作的合作者创造一个共享的空间。会议聚集和会议控制为合作者创造了一个“见面”的**共享会议空间**，而参加会议的合作者还需要有共同完成某一项工作的**共享工作空间**。共享工作空间又可以进一步划分为共享的操作空间和共享的信息空间，前者指一种允许工作组内任何成员对同一应用进行操作的机制，如白板、共享的编辑器等；后者是一种允许组成员创建、存储、浏览和提取共享多媒体信息的系统，如多用户的多媒体数据库系统。组成员可以通过后者将会议过程记录进数据库，也可以在会议期间提取出库中的资料供所有成员讨论。由于前者是实现协同计算的至关重要的机制，我们将在下节中给予专门的介绍。

8.5.3 应用共享控制

应用共享意味着每个组成员都可以对共享的应用程序（如编辑器）进行操作，而当其中某个成员进行操作之后，程序要将共享对象（如被编辑的文件）执行该操作的结果反映给每一个组成员。通常，共享对象在共享窗口中显示，成员甲对共享对象进行操作的结果不但更新自己的共享窗口中的内容，也对其他成员的共享窗口中的内容做同样的更新。

1. 共享工作空间的结构

共享工作空间的结构与会议控制方式相类似，也存在集中式和分布式两种。它们之间的不同之处是，在这里集中和分散是针对数据的分布而言的，而在会议控制中则是针对控制信息的走向而言的。

图8-36(a)给出了集中式的共享工作空间结构。应用程序的执行只在某一个成员处进行，任何一个组成员的操作指令都送到这个节点，而应用程序执行的结果（被某个操作改变了的共享对象）

被送给所有其他的组成员。集中方式的优点是容易维护，因为只有一份应用程序在对共享对象进行操作。同时，利用这种方式也容易将原来为单用户设计的应用程序改变为组工作方式。集中式结构的缺点是每一次操作后，都要将结果传送给所有组成员。对于图像类的对象，结果输出的数据量是很大的，需要占用大的通信带宽。此外，当进行操作的组成员和应用程序分处两地的话，从发出操作指令到操作结果返回的响应时间也比较长。

分布式或称为复制式结构如图 8-36(b)所示。在这种结构中，每一个组成员处都有一份应用程序的复制版本，一个成员发出的操作指令被送往所有成员处，各处的应用程序分别执行该操作，并将结果显示在本地共享窗口内。这种方式明显的优点是操作指令的数据量很小，占用的通信带宽少，而且操作在本地执行，因此结果返回到界面快、响应时间短。复制式结构的缺点是维护比较困难，因为程序在不同地点分别执行，难以保持数据的一致性，可能在经过一段时间的操作之后，各组成员在共享窗口中看到的内容已经不一样了。在复制式结构中，应该尽量避免使用取决于定时信号的操作(如按下鼠标使显示屏滚动)，因为这种操作会产生不确定性。此外，究竟在何时(预先、或在会话开始时、还是在需要时)和以什么样的方法将共享应用的复制版本分发到各处也是值得考虑的问题。

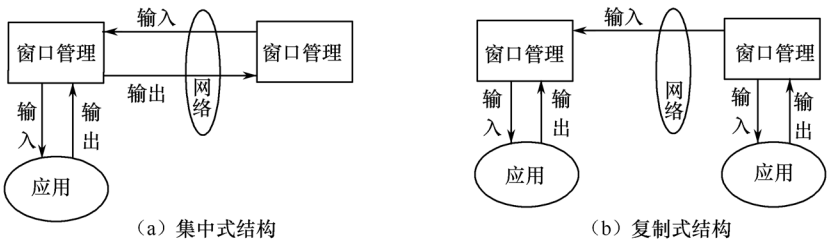


图 8-36　共享工作空间结构

2. 发言权控制

在共享工作空间中，每一个组成员都可以对共享的应用进行操作，因此实现应用共享所涉及的一个重要问题是并发控制，即解决组成员对共享空间访问的冲突。这也就是所谓的发言权(Floor)控制，只有持有发言权(或令牌)的组成员才能对共享对象进行操作。对发言权控制的讨论一般分为策略和机制两个方面。策略规定组成员请求、获得和释放发言权的方法，例如可以同时操作的用户数和操作的粒度(如访问字符，还是访问段落)等。策略需要通过相应的机制得以实现。

人们已经提出多种发言权控制的机制，如集中加锁、令牌传递和依赖性检测等[4]。最简单的发言权控制机制是在任何给定时刻只有一个组成员拥有发言权，在其他成员请求时，该组成员才把发言权传递给另一个发出请求的组成员。

3. 工作组界面

如何有效地显示共享的工作空间也是应用共享中需要考虑的问题。通常你所见即我所见 WYSIWIS(What-You-See-Is-What-I-See)是设计组界面时遵循的基本原则，因为工作组的所有成员应当同时看到某个操作的结果，否则组成员就不能“平等地”对共享空间中所发生的事件做出反应。但在另一方面，组成员也希望有一部分私有的空间，这导致了在一些系统的界面中出现公共和私有窗口之分，称之为松弛的 WYSIWIS，或者你所见不是我所见 WYSINWIS(What-You-See-Is-Not-What-I-See)。此时，保持组成员公共窗口内容的一致性就具有更重要的意义。

在工作组界面中用多种方法标识其他组成员的操作结果对于提高工作效率是有益的。例如，用黄、红、蓝、黑色字体分别表示文件中新近的和不是新近的以及未作修改的结果；又如在公有窗口中用阴影标出某段文字，表示此时有一个组成员正在自己的私有窗口推敲、修改该段内容，其他组

成员就会去考虑其他段落，等等。在同步会议系统中，也可以通过语音交谈来协调组成员对共享空间的操作。

8.6 多媒体流式应用系统与终端

8.6.1 流媒体的概念

在娱乐领域中应用的多媒体系统通常具有服务器——客户端的结构，如节目中心通过服务器向用户（客户端）进行多媒体内容的广播、点播和直播服务。其具体的做法是，将包含视频和音频信息的节目预先录制下来，存储在服务器硬盘或光盘等介质中；然后节目中心按预定的时间表从服务器中提取出节目通过有线或无线网络向用户进行**广播**，或按照用户（或用户组）提出的请求向其传送特定的节目，后者称为**点播**。所谓**直播**，则是指对视/音频信号进行现场采集和压缩，然后传送到节目中心的服务器，经服务器再向用户进行广播。与视听通信系统收发两端对等不同，这类多媒体系统的结构是不对称的，服务器和用户终端在复杂性和处理能力上有很大差异，上行（用户终端到服务器）和下行（服务器到用户终端）信道的带宽通常也是不同的。由于在这类多媒体系统中，视/音频信息通常是预先采集、压缩和存储好的，因此此类应用称为**存储视频**（Stored video）应用。与此相对应，视听系统中视、音频信号边采集、边压缩、边传输到接收端的情况（如多媒体会议），则称为**实时视频**（live video）应用。

在存储视频应用中，一种常见的做法是将存储的视频数据以文件形式通过因特网全部**下载**到用户终端，然后终端通过读取本地文件实现视频的连续播放。这种方式可以避免传输带宽不足或延时抖动过大对播放质量的影响，但是由于视、音频文件一般都很大，因此需要用户端有较大的存储容量；同时带宽越窄，下载时间越长，用户等待的时间就越长，有时甚至需要等待数十分钟。如果用户端在接收到文件前面一部分数据时，例如几秒或几十秒后就播放，与此同时，服务器不间断地向用户提供后续的数据，则可以大幅度减少用户等待的时间。这种边传输、边播放的方式称为**“流媒体”**方式。流媒体服务器可以单播、多播或广播的方式向用户提供服务。对于实时视频应用（如可视电话）而言，视/音频数据不言而喻总是边传输、边播放、以流的方式工作的，因此流媒体的概念只在存储视频应用中被强调。

如前所述，流媒体是在因特网上采用流式传输方式工作的音视频应用。要获得良好的播放质量，流媒体方式对传输网络的带宽、延时和错误率有相当高的要求，而当前的 IP 网不提供完善的 QoS 的保障机制，同时也不能在满足用户不同 QoE 要求的条件下有效地支持多播，因此设计 IP 网上多媒体流式传输系统和协议面临着许多挑战性的问题。

8.6.2 多媒体流式应用系统的结构

1. 顺序式流传输

最初的 Web 页面只包括文字和图像，当用户选择某个页面时，包含这个页面的文件首先从 Web 服务器下载到用户终端，然后终端的浏览器将页面呈现在显示器屏幕上。如果页面中包含有视、音频片断，下载时间会较长。图 8-37 给出利用 Web 服务器传输视/音频的系统结构。当用户在网页上点击一个连接视、音频的超链时，浏览器将它当作一个文本文件那样处理，因此，浏览器的 HTTP 先与超链指明的服务器上的 HTTP 建立 TCP 连接，然后请求得到超链指明的文件。当浏览器接收服务器送来的文件时，从服务器应答消息的头部得知文件内容类型（如 Audio/MP3），浏览器就调用相应的播放器（如音频播放器），并把接收到的文件传送给播放器。这种方法的缺点是，先要把文件完全下载，再由浏览器将文件传送到播放器，当文件很大时，将引入不可接受的延时。

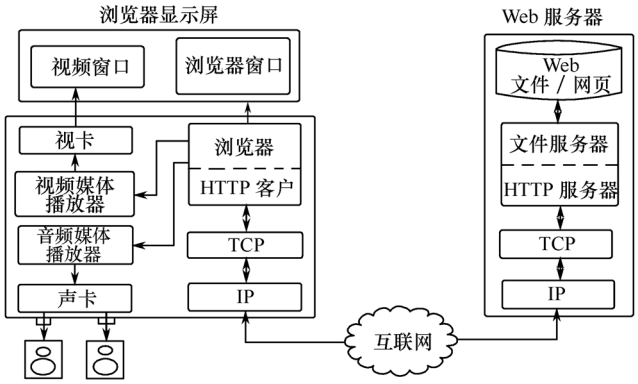


图 8-37　利用 Web 服务器传输视/音频文件的结构

图 8-38 给出一种改进方案。采用这种方案时，每一个视/音频文件都伴随有一个**元文件**(Meta file)。元文件中包含它所对应的视/音频文件的 URL 地址，以及文件内容类型等描述信息。同时，网页超链所指的地址是元文件 URL，而不是视/音频文件本身的 URL。因此，当用户点击超链时(GET 请求)，服务器返回的 GET 应答消息中包含元文件的内容。浏览器像在图 8-37 所示系统那样通过返回消息得知文件内容类型，调用相应的播放器；不同的是将元文件的内容(而不是视/音频文件)传送给播放器。播放器判断接收到的是元文件，则从中读出视/音频文件的 URL，继续按通常的 HTTP/TCP 协议获取视/音频文件的内容。播放器开始接收到视/音频数据后直接将它们放进播放缓存器，在规定的起始延时之后，播放器从缓存器中取出数据进行解压缩和播放。这种方法消除了通过浏览器获得视/音频文件所引起的时延，并能在结束文件传输前就开始播放，因而是当前使用 Web 服务器存储和传输视/音频时普遍采用的方法，也称为**顺序流式传输**(Progressive streaming)。在这种方法中视/音频文件仍然是通过 HTTP/TCP 传输的，我们知道 TCP 能保证传输的无差错性，故而视/音频的传输质量较好，但在网络可用带宽不足时，它的重传机制引入大的延时。一个粗略的估计是，在可用的 TCP 吞吐量大于 2 倍的媒体流速率时，TCP 可以提供好的流传输[5]。由于因特网的带宽很不稳定，因此此种方式较适合于传输短视频。在传输长视频时，为了防止延时过大造成的播放中断，播放需要等缓存器中有足够多的数据后才开始，用户需要等待的起始延时仍然较长。同时，在这种系统中，没有对视/音频文件的传输进行控制的机制，因此只能顺序播放，而不能进行快进、快退等操作，这也是顺序流式传输这个名称的由来。一些商用的流媒体系统采用了这种结构。

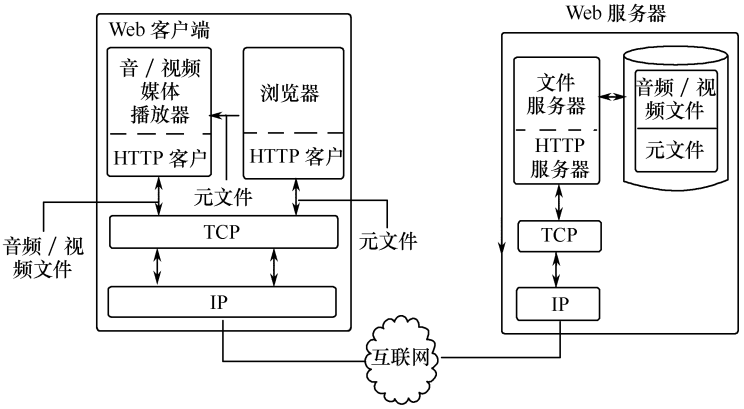


图 8-38　利用 Web 服务器和元文件传输视/音频流的结构

2. 交互式流传输

图 8-39 给出在因特网(IP 网)上进行视/音频流式传输的典型系统结构,这也是许多流媒体应用、IP 网视频点播和 IP TV 等系统采用的结构。图中在服务器端,除 Web 服务器外还有一个流服务器,用来存储和输出视/音频文件。由于视频比音频的数据量大得多,**流服务器**也称为媒体服务器,或视频服务器。在客户端,媒体播放器则分成了数据和控制两个部分,其数据部分负责媒体数据的接收、解压缩和显示等,控制部分则负责产生控制命令对播放过程进行控制,如快进、快退、暂停等。图中同时标出了各对连接中使用的通信协议,其中包括节目浏览和选择时使用的 HTTP;控制流服务器操作的实时流协议 RTSP 和在媒体流传输时使用的 RTP 协议。

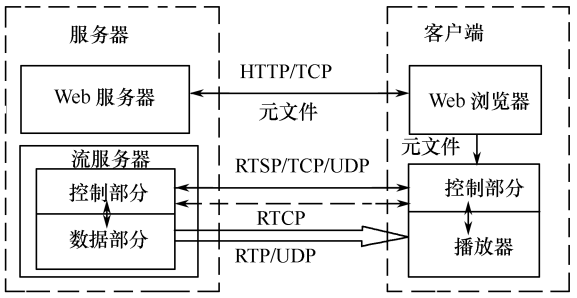


图 8-39　视/音频流式传输结构

图 8-40 表示出图 8-39 所示系统在工作时控制信息的交互过程,其具体步骤如下:

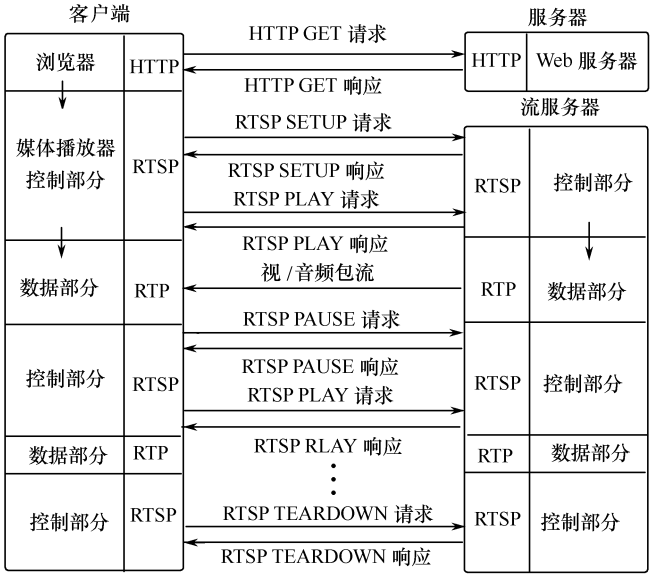


图 8-40　流式传输的命令交互过程

(1) 用户通过网页点击自己想看的节目,浏览器向所选超链的 URL 指明的 Web 服务器发出 GET 请求,索要相关的元文件。

(2) Web 服务器向浏览器返回 GET 响应消息,其中包括元文件内容。

(3) 浏览器根据返回消息得知文件内容类型,然后选择相应的媒体播放器(音频或视频),把元文件内容传送给该播放器的控制部分。

(4) 播放器控制部分从元文件内容中读出媒体文件的 URL,并向具有该 URL 的流服务器发送 RTSP SETUP 请求消息,申请建立 RTSP 会话。在这一过程中,还要传送一系列的参数,例如

媒体文件名、RTSP 会话号、要求操作的模式、RTP 端口号和认证信息等。如果这些都是可接受的，流服务器的控制部分返回 RTSP SETUP 应答消息，其中包括由服务器分配的唯一的 RTSP 会话标识符。该会话标识符在此后的所有 RTSP 请求消息中使用。

(5) 当用户按下播放器的“播放”按钮时，播放器控制部分向流服务器控制部分发送 RTSP PLAY 请求消息(包括会话标识符)。流服务器根据会话标识符知道读取权限已经认证，则向播放器控制部分返回确认消息。播放器收到确认消息后做接收数据流的准备。

(6) 服务器控制部分启动数据部分的媒体流读取和传输机制，向播放器在会话建立请求消息中给出的 RTP 端口传输媒体流。

(7) 播放器收到媒体流后，将数据放入播放缓存器；经过预定的起始延时后，播放器开始从缓存器中读取数据，并进行解压缩和显示。

(8) 播放过程中用户可能要做暂停和快进、快退等操作，这些操作命令通过相应的 RTSP 请求消息传送到服务器控制部分，使服务器进行相应的操作。

(9) 当用户按“结束”按钮时，播放器控制部分向服务器控制部分发出 RTSP TEARDOWN 请求；服务器返回响应消息后，此次 RTSP 会话结束。

从上面的工作过程我们可以看出，播放器从接收到媒体数据到开始播放也要有一个延时，其大小取决于网络传输的延时抖动，一般是较小的(几秒)。在这种系统中，视/音频数据采用 UDP 传输，网络带宽和错误率对视/音频传输的质量产生直接的影响。此外，RTSP 控制协议的引入使得对播放过程进行录像机式的操作(暂停、快进等)成为可能，在下一节中我们将对该协议作进一步的讨论。

8.6.3 实时流协议(RTSP)

实时流协议(Real-Time Streaming Protocol，RTSP)是一个应用层协议，在 IETF RFC 2326 中定义。RTSP 的目标是建立对一个或多个实时媒体流的控制，在典型应用中它不负责实时媒体本身的传输；实时媒体流通过其他协议(默认为 RTP)传输，RTSP 的作用是对流服务器进行“网络远程控制”，以实现与录像机相似的控制功能。

RTSP 相对于流服务器的作用类似于 HTTP 相对于 Web 服务器，因此 RTSP 在语法和操作上采用与 HTTP 相似的方法，这使得 HTTP 的扩展机制在大多数情况下都可以加进 RTSP。同时，RTSP 报文可以由标准的 HTTP 或 **MIME**(Multipurpose Internet Mail Extention)解析器解析。RTSP 可以重用 HTTP 的缓存、代理及安全机制。但是 RTSP 也与 HTTP 有重要的区别，例如，(1)RTSP 与它所控制的媒体数据可以通过不同协议和不同通道传送(如 RTSP 通过 TCP，数据通过 RTP/UDP)，换句话说，媒体数据是在“带外”传送的。(2)HTTP 是一个不对称的协议，客户端发出请求，服务器应答；而 RTSP 是对称的，客户端和服务器均可发出请求。(3)HTTP 是一个无状态协议，即发送一个命令后，连接会断开，且命令之间没有依赖性；而 RTSP 是有状态的，以便在请求确认后很长时间内，仍可设置参数控制媒体流。

通过对图 8-39 所示系统的讨论，我们已经对 RTSP 的工作过程有了基本的了解。对 TCP 和 UDP 来说，RTSP 服务器的默认端口都是 554。在 RTSP 中没有“RTSP 连接”、只有“RTSP 会话”的概念。在一个 RTSP 会话中，用户端可以打开或关闭多个可靠传输层连接(如 TCP)，也可以使用无连接协议(如 UDP)向服务器传送 RTSP 请求消息。在建立 RTSP 会话之前，客户端需要以某种方式(在图 8-40 的系统中是通过 HTTP)获得一个播放**描述文件**(即元文件)。对该文件的格式 RTSP 没有做限制，例如描述文件可以采用 SDP 表示，但该文件应包括此播放过程有几个媒体流、每个流的参数(如编码方式等)、存储每个流的文件名和流服务器的 URL(同一播放过程的不同流，如音频和视频，可以存放在不同的服务器上)，以及使用的协议栈等内容。客户端和服务器也可以

通过直接交换 RTSP DISCRIBE 请求和响应消息来获得媒体描述信息。针对每一条媒体流，客户端和服务器都需要通过交换 RTSP SETUP 请求和响应消息来建立一个 RTSP 会话。

RTSP 消息由命令行及参数、若干头字段和消息体构成。RTSP 消息是文本形式的，当增加新参数时有自说明的作用。由于消息中的参数不多，再加上使用 RTSP 命令的频率很低，因此文本式消息处理效率较低的缺陷在这里并不是问题。

最后需要指出，RTSP 不仅支持单播，也支持由服务器选择或由用户选择地址的多播。RTSP 不仅能如图 8-40 所示那样控制服务器提取和传输媒体流的过程，还支持“邀请”一个服务器加入到正在进行的会议中来播放或记录共享的文件，会议参与者均可对该服务器进行远程操纵；以及支持服务器通知用户有新的媒体流加入到正在进行的播放过程中，这在实况转播的情况下十分有用。

8.6.4 流媒体文件

在图 8-39 所示的系统中，流服务器需要从文件中读出媒体数据，然后按照协议要求将数据封装成包（如 RTP/UDP 包），再通过 IP 网络输送给用户。图 8-41 表示出这个读取、封装和发送的过程。由于对每一个用户都需要启动这样一个过程，当流服务器向数百个到数千个用户提供服务时，提高每一过程的效率就显然尤为重要。因此在流式应用系统中存储媒体数据的文件（称为**流媒体文件**）通常与用于本地播放的文件（如 AVI 等）在格式上有所不同，它的格式应该能够支持边读取、边播放，并且能够方便服务器对数据的封装，以提高流服务器的工作效率。值得注意，存储已编码的媒体数据的文件格式与媒体数据的编码格式（如 H.264）是两个不同的概念。文件格式规定的是数据的存放方法，不同编码格式的数据，可以以同一种文件格式存放。

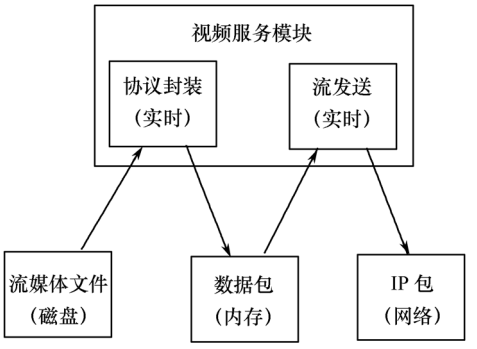


图 8-41　流服务器的主要业务流程

一个流媒体文件通常包含媒体数据本身、描述媒体数据的元数据和索引三个部分的内容。按时间顺序排列的媒体数据序列构成“**轨道**”（Track）；不同的媒体（如音频和视频）放在不同的轨道上。为了服务器提取数据的方便，媒体数据通常按相等的播放时间间隔分割成段，而具有流间同步关系的媒体数据段（如音频和视频）常常交错排放在一起，这使得流间同步在发送过程中容易得到保证。**元数据**描述文件的全局信息（如文件长度、播放时间、轨道类型和数目等）和局部信息（如段的数据排放方式、与播放时间和/或 RTP 包的对应关系等）。全局描述信息一般出现在文件的开始，而局部描述信息在不同的文件格式中可能放在文件开始或者数据段的开始部分，甚至单独用一个轨道来表述。**索引**给出每个轨道各数据段（或时间点）的数据在文件中的偏移量，以便于实现节目播放时的拖放功能和对特定时间点的搜索。

流式应用系统众多的用户终端可能是**异构**的，即具有不同的图像分辨率、处理能力和接入带宽。为了给异构的用户群提供服务，最简单的方法是将同一节目压缩成具有不同码率的几个版本，服务器根据用户的需要选择对应的版本进行输出，例如对使用手机和电缆调制解调器的用户可以分别传送 128 kb/s 和 512 kb/s 的码流。当服务器拥有同一节目不同速率的几个版本时，还可以在节目传送过程中改变输出速率（从不同的版本中读取数据），以适应网络带宽的动态变化。因为当带宽下降时，传送一个与网络带宽相适应的低码率的码流，在接收端获得较低质量的完整图像，与传送高于带宽的高码率码流造成大量丢包、在接收端得不到完整图像的情况相比，用户的主观感受要好得多。为了支持如上应用，通常将精心排放的同一节目的几个不同码率的版本放在同一个流媒体文件的若干个“轨道”上，此时生成的文件称为**多速率文件**。

值得注意，改变码流速率时轨道间的切换需要在I帧处进行，以防止在新流的起始处发生帧间预测参考帧的缺失。因此，要想在网络带宽发生变化时及时地进行流的切换，就需要减小相邻I帧之间的距离，这将引起编码效率的降低。为了克服这个缺点，在H.264中引入了**SP帧**的概念。SP帧是特殊编码的P帧，在该帧进行流切换不会引起新流起始处参考帧的缺失。有兴趣对此做进一步了解的读者可参考文献[6]。顺便指出，与SP帧的作用类似，H.264中还引入**SI帧**用于不同节目流之间的切换，这在有些应用场合，例如IPTV中变换频道时是十分有用的。

具有代表性的流媒体文件包括MOV、3GP、RM、RMVB、ASF、FlV和MP4等。ISO/IEC 14496－14规定的MP4是一种灵活和广泛使用的视音频数据存储文件格式。它支持本地播放和流式应用。下面我们从流媒体文件的角度对它进行简单介绍。MP4文件的基本单元是称为“**盒**”(box)的容器。每个盒具有图8-42(a)所示的结构。盒可以层层嵌套，最外层的盒主要有ftyp、moov和mdat三种类型。ftyp一般出现在所有盒之前，描述文件的全局元信息；moov中包含描述各条轨道的元信息；而mdat中则是音、视频媒体数据。图8-42(b)给出一个简化的MP4文件结构，其中表示出moov和mdat与它们内部的主要子盒之间的嵌套关系。

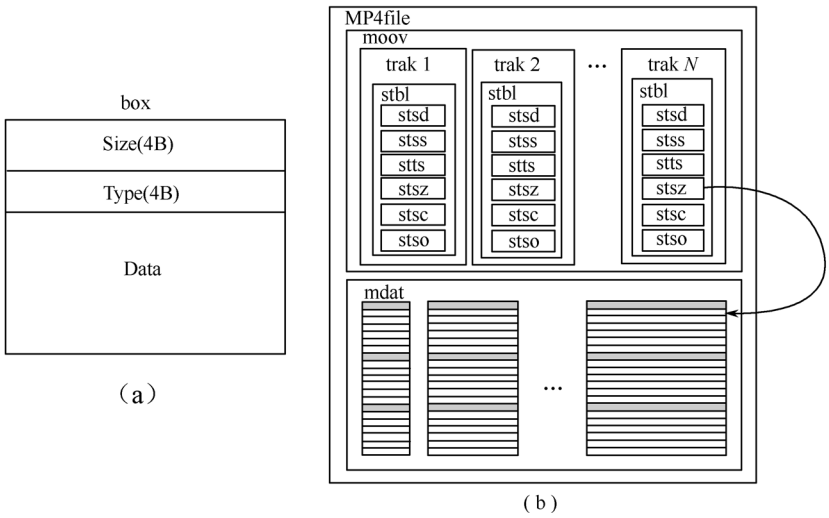


图8-42　简化的MP4文件结构

不同速率/编码方式的视、音频序列，或不同语言的音频序列分别放在不同的轨道上。在mdat中，媒体数据以**采样**(sample)为单位存储，视频的一个采样通常为一帧数据，音频的一个采样则为一个或多个音频样值。若干个采样组成一个**块**(chunk)，通常对应1秒左右的数据。在一个块中，采样按播放时间顺序排列连续存放；而块的排列可以不是顺序的。在流式应用中，往往将同时播放的音视频块交错排列顺序存放，以利于服务器的提取。在moov中，每一个轨道由对应的track盒中一组子盒进行描述，其中最重要的是采样表盒stbl[见图8-42(b)]。stbl包含6个重要的成员，其中stsd说明采样的编码方式和配置；stss指出可以随机接入或进行码流(轨道)切换的帧(也称关键帧或同步帧)的位置[图8-42(b)mdat中灰色条]；stts描述每个采样的显示时长，通过对stts的解析可以得到采样与播放时间之间的映射关系；stsz和stsc分别给出每个采样的大小和每个块含有的采样数；stso指出每个块在文件中的偏移量(字节数)。可以看出，服务器或客户端播放器通过以上信息很容易搜索到所需要的数据。

在流媒体应用中，MP4还可以包括索引轨道，它在moov中也有相应的描述信息。音、视频分别有自己的索引轨道，索引轨道的采样也放在mdat中。索引采样含有音视频数据传输层打包的信息，如一个视频采样对应的RTP包数、包头、序列号等。流服务器根据索引轨道描述信息找到对应的索引采样，然后按索引采样给出的信息对音、视频采样进行RTP打包和传送。

8.6.5 流服务器

流服务器是图 8-39 系统中的重要设备，首先它需要强调操作系统的实时性。通常我们在工业控制机中也常常采用实时操作系统，但在那里操作系统的实时性体现在对(事件)中断的响应上；而在流服务器中则主要体现在对媒体流的处理过程中保证流的连续性。虽然传统的操作系统和文件系统并未对流服务器的设计提供必要的支持，但由于目前磁盘的数据传输速率远远高于一个连续媒体流的码率，如 EIDE 硬盘的数据传输速率为 16.6 MB/s，Ultra-SCSl 硬盘可达 40 MB/s，而符合 MPEG-1 标准的视频流的码率为1.5 MB/s，未经压缩的具有 CD 质量音频流的码率大约为 0.2 MB/s，因此，只要采用恰当的技术，流服务器可以同时完成多条连续码流的实时输出。而由多个服务器构成的服务器阵列则可以提供更多的码流输出。涉及流服务的技术包括与操作系统有关的任务调度、资源管理、磁盘调度，和与存储子系统有关的数据在物理介质上的排放方式以及层次化存储结构等。下面我们分别予以讨论。

1. 任务调度

当为许多用户服务时，流服务器按照一定的策略(称为任务调度)快速地把第一个用户在一段时间内所需要的数据取出来放在缓存区 1 内(见图 8-43)，紧接着把第二个用户在这段时间内所需要的数据提取出来放在缓存器 2 内，等等；与此同时，服务器的输出线程按照用户播放速率的要求把每个缓存区中的数据轮流放上线路，使线路上的每个流都是连续不间断的。同时，不等缓存区中的数据用完，服务器又把每个用户在下一段时间所需要的数据提取出来放入缓存区。因此流服务器为一个用户服务的过程可以被抽象为一个由若干带有时限的任务构成的序列，其中每项任务负责从存储介质中读取一定数量的数据并传送给用户，每项任务的时限与该段数据的预定播放时刻相对应。由于连续媒体的播放时刻具有明显的周期性，各项任务的时限也呈现出周期性的特点。

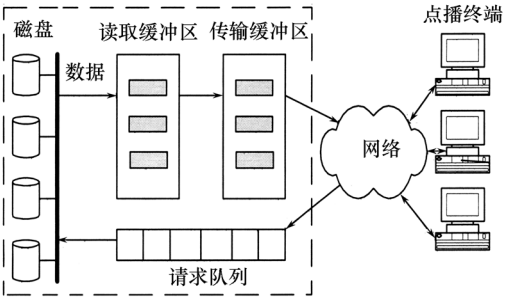


图 8-43 流服务器的 I/O 模型

图 8-44 给出了使用两种常用算法 **EDF**(Earliest Dead-line First)和**速率单调调度**(Rate Monotonic Scheduling)同时为一个高速码流和一个低速码流服务时进行任务调度的例子。在 EDF 中，时限越早的任务越早执行。由图看出，高速流的任务 1 和低速流的任务 A 首先被执行，然后执行时限较迟的高速流任务 2。在速率单调调度算法中，任务按其所要求的速率预先设定优先级，按优先级决定执行的顺序。图中低速流的任务 A 在高速流的任务 1 之后执行，在 A 未完成之前出现高速流 2，由于任务 2 的优先级高，因此先执行 2，而剩余的任务 A 在任务 2 完成后执行。一般来说，速率单调算法在 CPU 利用率小于 69%时可以满足所有任务的时限[7]；而 EDF 能达到 100%的 CPU 利用率，但在过负荷情况下有一些任务的时限可能得不到满足。

2. 资源管理

资源管理的主要功能为**接纳控制**和**资源分配**。接纳控制模块负责在新用户到来时，判断能否在继续满足现有各用户服务要求的前提条件下为新的用户提供服务；资源分配模块则在接纳新用户时为其预留资源。

接纳控制根据系统当前的剩余资源以及申请所需的处理来判断是否为该用户提供服务。系统的剩余资源通常由 CPU 循环周期内的剩余时间(如图 8-44 所示)来标度，而对于申请的处理则需要根据用户所要求的 QoS 级别以不同的方式来确定。具体地讲，当用户要求得到确定性保障时，

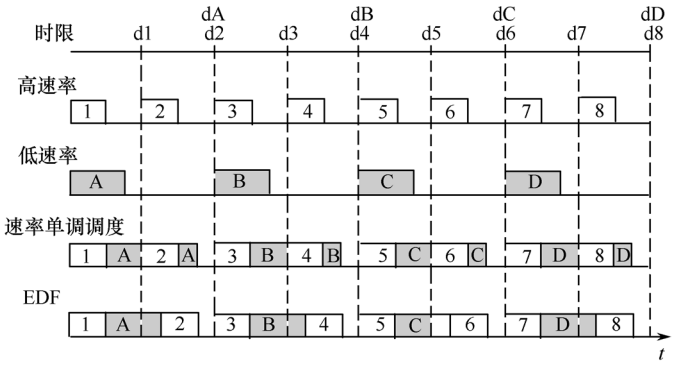


图 8-44　任务调度的两种算法

处理开销由最坏情况下处理一个申请所花费的时间来表示；而当用户申请统计保障时，处理开销的确定则有赖于流服务器对处理申请所花时间的统计规律的分析。应当说，由于申请的处理时间会受到磁盘调度方式、连续媒体数据在磁盘上的排放方式以及连续媒体码率的波动情况等诸多因素的影响，所以对处理时间变化规律的分析往往十分困难，而一旦流服务器确定了处理时间的统计特征，接纳控制则可利用处理时间的某种统计平均值来表示申请的处理开销。当用户要求“尽力而为”服务时，流服务器对能否满足一个用户申请的时限不做任何承诺。此时，服务器可以通过采取一些措施（如降低连续媒体的质量等）来降低处理申请所花费的开销，以便尽可能满足更多申请的时限，而客户端也可通过加大等待时间等方法来尽力保障连续媒体的正常播放。显然，以最坏情况下申请的处理时间来判断是否接纳新用户，系统资源利用率较低，也就是说，当流服务器提供 QoS 的确定性保障时，它所能同时服务的用户数目比提供统计性保障时要少。

与接纳控制相对应，流服务器的资源分配也分为确定性的和统计性的。确定性资源分配机制按最坏情况为接纳的新用户预留资源；而统计性资源分配机制则允许小概率的 QoS 降低以换取资源的更高利用率。

3. 磁盘调度

当连续媒体数据在磁盘上按一般离散媒体数据的方式存储时，我们通常利用缓存器和特殊的磁盘调度算法来减少数据读取的延时抖动，以保障媒体流的连续性。

在磁盘调度机制接收并处理来自多个进程的磁盘读或写请求时，处理一个请求所花费的时间由以下因素决定：

寻道时间　即磁盘的读/写头由当前位置移动到指定磁道所花费的时间。

旋转延时　即磁盘的读/写头移动到指定磁道后，相应的扇区旋转至读/写头位置所花费的时间。

数据传输速率　即单位时间内所能读或写的数据量。

上述三个因素中，寻道时间对处理一个请求所需时间的影响最大，因此传统的磁盘调度算法主要考虑降低寻道时间和公平地为各进程提供服务。常见的传统磁盘调度算法有 FCFS 和 SCAN 等。

FCFS（First-Come-First-Served）算法按照读/写请求到达的先后次序提供服务。这种算法的公平性是最佳的，但寻道时间较长。它的工作过程由图 8-45 所示，其中的数字表示请求数据所在的磁道。图中首先有一个要求 12 磁道数据的请求到达并进入等待队列；根据先到先服务的准则，磁盘调度机制为队列头的 22 磁道请求服务。然后，随后到达的 45 磁道的请求加入到剩余队列（40、12）之尾；队列最前面的 40 磁道请求得到服务；等等。

另一种常见的传统算法是 **SCAN**，它以当前磁头的移动方向为依据来选择下一个服务请求，将要被服务的请求不会变更磁头当前的移动方向；当被缓冲的所有请求都不能满足条件时，SCAN 算法将反向移动磁头，并以新的移动方向选择将被处理的请求。SCAN 算法的工作过程由图 8-46 来表示。在图示的第 1 个请求(22 磁道)被服务之后，磁头从第 22 磁道向磁道数增加的方向移动，等待队列(第 2 个队列)中磁头首先移动到的磁道的请求(40 磁道)被服务，以此类推。当第 50 磁道的请求被服务之后，磁头已移动到磁盘的边沿，此时等待队列中的请求(12，16，30)均在小于 50 的磁道上，因此磁头反向移动并对 30 磁道的请求给予服务，然后再对 24 磁道的请求服务。显然，与 FCFS 相比 SCAN 算法有较低的平均寻道时间，并保持了较好的公平性。

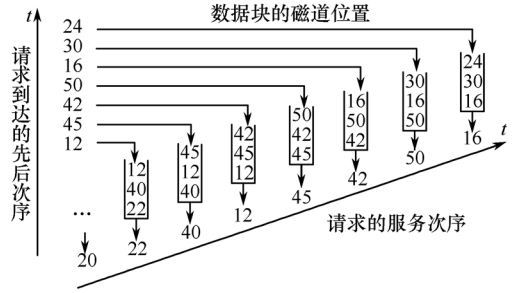


图 8-45　FCFS 算法的工作过程

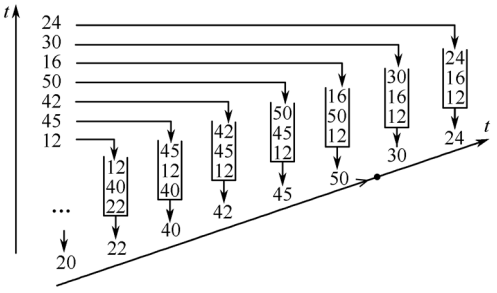


图 8-46　SCAN 算法的工作过程

对 SCAN 的一种改进产生了 **C-SCAN**(circular SCAN)算法。其主要不同之处在于，当磁头沿某一方向移至磁盘边沿时，C-SCAN 算法并不像 SCAN 算法那样变更磁头的移动方向，而是将磁头空移至另一边沿，然后磁头仍以原方向移动来处理请求。尽管 C-SCAN 算法在降低平均寻道时间方面逊于 SCAN 算法，但在公平性方面却优于 SCAN 算法。

由于上述传统的磁盘调度算法以降低寻道时间、提高磁盘的数据吞吐率为主要目的，未考虑各读/写请求的完成时限等实时因素，所以直接用于连续媒体数据的提取有不足之处。

我们在实时任务的调度中介绍过一种 EDF 算法，它是用于连续媒体的比较好的调度算法，但是将 EDF 算法用于磁盘调度的话，它在调度过程中只考虑了各请求的完成时限，而没有考虑数据在磁盘上所处的位置。将 EDF 和 SCAN 相结合，人们提出了 **SCAN-EDF** 算法。它的调度机制是，首先依照 EDF 策略为那些具有最早完成时限的读/写请求提供服务；当若干读/写请求具有相同的时限时，调度机制以 SCAN 方式为这些请求提供服务，其工作过程如图 8-47 所示。图中矩形框左侧的数字表示时限，右侧的数字表示请求数据所在的磁道。由 SCAN-EDF 的原理不难看出，具有相同时限的申请的数目越大，平均寻道时间越小，磁盘吞吐率越高。

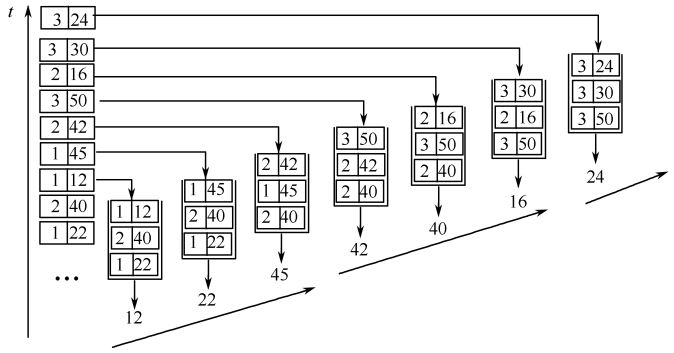


图 8-47　SCAN-EDF 算法的工作过程

另一种用于实时媒体的磁盘调度算法是 **GSS**(Group Seping Scheduling)。GSS 算法首先依据

完成时限的近似程度将各请求分组并为每组请求分配一个统一的时限；随后，GSS 根据组时限依次为各组请求提供服务。对组内各请求，GSS 采用 SCAN 方式进行磁盘调度。GSS 算法的工作过程可由图 8-48 表示，图中所有请求按时限（磁道数左面的数字）划分成 3 个组。

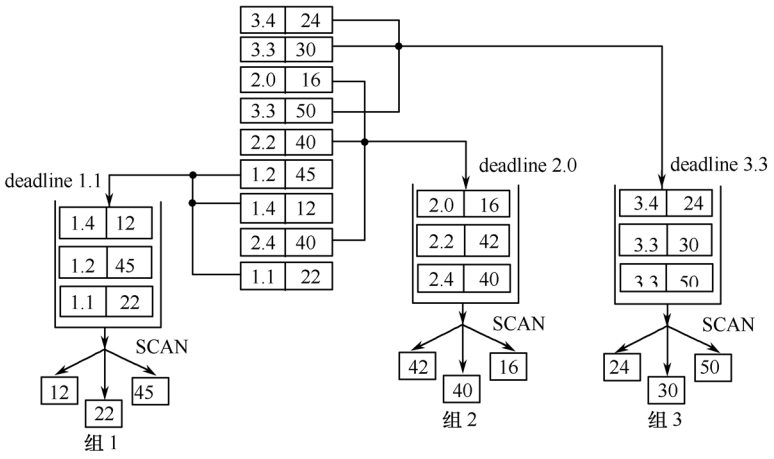


图 8-48　GSS 算法的工作过程

由于一个用户的各次请求的时限具有周期性的特点，所以在各循环内，分组状况以及各分组的位置均不会发生变化，这时只需缓存一个服务周期的数据就可以保证媒体流不间断。同时，由于较多的请求具有了相同的时限且以 SCAN 方式进行调度，所以磁盘吞吐率和所能同时服务的用户数目比采用 SCAN-EDF 时有较大的增长。

4. 媒体数据的排放方式

除了调度算法之外，媒体数据的排放方式也是影响存储子系统吞吐量的一个重要因素。如果一个视频文件全部存储在一个磁盘上，那么存取这个文件的并发用户数，也就是可以同时观看这个视频节目的用户数，由该磁盘的最大吞吐量所限制。为了突破这个限制，人们提出了**条带**（strip）存储方法。在这种方法中，一个多媒体文件分段存储在若干个磁盘上，这些磁盘可以并行地进行存取，其结构如图 8-49 所示。由于分布在不同磁盘上的数据块可以并行地存取，这就成倍地提高了磁盘系统（磁盘阵列）的吞吐量。在条带存储的设计中，一个重要的问题是确定单个磁盘上存储数据块的大小和一个文件跨越的磁盘个数，以使磁盘的负载尽量均衡和文件的提取时间尽可能地小[8]。

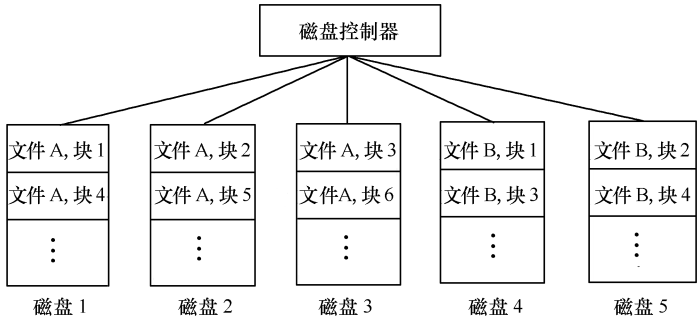


图 8-49　数据的条带存储

5. 层次化存储结构

流服务器（或服务器阵列）管理着为数众多的连续媒体文件，对物理存储介质的容量有着很高

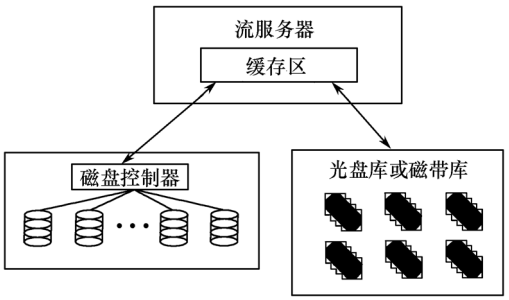


图 8-50　层次化存储结构

的要求；另外，它需要同时为多个用户实时地提供多个连续码流，对存储介质的数据传输速率也有很高的要求。如何同时满足容量及数据传输速率两个方面的要求是一个必须解决的问题，如图 8-50 所示的层次化存储是一种较为理想的解决方案。

通常 RAM 称为层次化存储结构的第 1 层，为流服务器提供实时读/写连续媒体数据所需要的各种缓冲区。各种形式的磁盘（如 SCSI 磁盘、IDE 磁盘等）和磁盘阵列（如 RAID）为第 2 层，用来存储对传输速率要求高、频繁被实时调用的数据，常称作**在线**（On-Line）存储设备。由于磁盘容量相对较小、价格较高，因此光存储介质（如各种光盘库）或数据磁带常作为第 3 层，用来存储不经常调用的数据，称作**近线**（Near-Line）存储设备。近线存储设备有适中的数据传输速率、容量及价格，是在容量方面对在线存储设备的一种补充。

层次化存储系统设计的一个重要问题是媒体数据在各层存储设备之间的迁移策略。迁移由连续媒体对象被访问的频繁程度和存储介质上可用空间的大小所决定。当一个连续媒体对象被访问时，为了保证数据传输速率满足要求需将该对象迁移到在线设备中。如果在线设备的可用空间降低到预先设定的阈值以外，那些长期未被访问的对象应该向下迁移到近线设备中。

6. 存储网络

在构建大规模的存储系统时常常需要利用网络将海量的存储设备与其他设备（如多台主机，服务器或服务器阵列）相连。存储区域网络 **SAN**（Storage Area Network）和附属于网络的存储结构 **NAS**（Network Attached Storage）是两种传统的存储网络。随着因特网技术的发展与普及，人们开始考虑基于 IP 来建立存储网络。**iSCSI** 就是基于 IP 建立存储网络的一种接口协议。服务器通过 **iSCSI** 将 SCSI 读写命令打在 IP 包中传送给本地 IP 网上的存储设备，来进行数据块的存取。近年来随着云技术的发展，云存储也是实现存储网络的不错选择。

8.6.6　内容分发网络

1. 内容分发网络的结构

媒体点播与广播或 IPTV 的用户通常分布在广阔的地域上，由中心服务器直接向这些用户提供服务需要占用大量的长途通信资源。比较经济的做法是将节目复制或通过长途线路传送到靠近用户的小区服务器，再由小区服务器向本区的用户提供服务。这就需要在 IP 网上构架一个**内容分发网络 CDN**（Content Distribution Network），将中心服务器上的节目传送到各边缘代理（小区）服务器。

内容分发网络 CDN 也称为内容传送网络（Content Delivery Networks）是构架在 IP 网上的一种**叠加**（overlay）网络，它由分布在广阔地域范围上的一组 CDN 服务器组成。CDN 原本针对 Web 内容的分发而提出，现在扩展到实时媒体的传送上。图 8-51 给出 CDN 结构的示意图。当用户向系统请求一个节目时，该请求被转发到离他最近的 CDN 服务器上，这称为用户请求的**重定向**。如果重定向所指向的服务器保存有用户所请求节目的复制版本，那么该服务器（也称**代理服务器**）向用户传递节目数据流；如果没有，则该服务器通过系统的**内容路由**功能找到和获取节目，然后传递给用户。由于用户由邻近的边缘代理服务器提供服务，因此降低了源服务器的负载和长途干线网的流量，节约了网络资源，使系统可以为更多的用户服务；同时由于用户和服务器距离较近，传输延时、延时抖动、丢包率相对较低，因而可以给用户提供较好的 QoS。

用户的重定向可以通过固定的配置来实现，即某一群用户由固定的服务器提供服务；也可以通

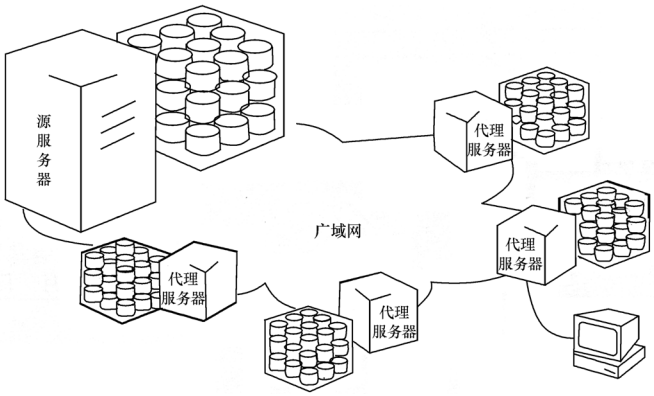


图 8-51　内容分发网络

过其他方法动态地实现，其中最常用的是基于域名服务(DNS)的方法。我们知道，DNS 的基本功能是建立域名和它对应的服务器的 IP 地址之间的映射关系。利用这一服务，用户的本地 DNS 服务器将用户所请求节目的 URL 转发给重定向 DNS 服务器，重定向 DNS 根据分发网络的 CDN 服务器分布情况将距离转发 DNS 最近的 CDN 服务器的 IP 地址返回给转发 DNS，该 DNS 再将此 IP 地址传送给用户。由于返回的实际是距离用户本地 DNS 服务器最近的 CDN 服务器的地址，因此在使用本方法时，用户必须邻近本地 DNS 才能真正获得“就近”的服务。

通过 CDN 提供的业务可以分为两大类：点播和直播。对于点播业务，如果边缘服务器保存了源服务器的所有节目，那么对于用户的所有请求都可以立即给予服务，但需要占用庞大的存储空间；而如果边缘服务器只在用户提出请求的时候才向源服务器索取节目，则会加大用户的等待时间和核心网络的流量。如何处理好这一矛盾，边缘代理服务器上应该存储哪一部分节目或节目的哪一部分，即它的**缓存策略**，是 CDN 涉及的一项重要技术。此外，我们注意到将海量的来自不同内容提供商的节目存储在一个源服务器(或服务器群)上是不现实的。当众多的节目分别存储在网络的多个节点上时，称为**合作缓存机制**(Cooperative cashing scheme)，此时如何组织 CDN 及如何找到所请求的节目(**内容路由**)是 CDN 中另一项重要技术。由于短期内在大范围内实现 IP 多播遇到不少困难，因此在 CDN 叠加网络上实现多播，称为**应用层多播**，是实现有效的内容分发的一个替代方案。对于直播业务，CDN 服务器将请求同一节目的用户会聚起来构成多播组，所有多播组的成员在 CDN 上构成多播树。当一个 CDN 服务器收到用户对该节目的请求、而它尚不是多播组的成员时，它通过内容路由将自己加入进多播树。应用层多播是 CDN 中又一项重要技术。下面我们将对上述几种技术分别进行简要的讨论。

2. 代理服务器的缓存策略

代理服务器的缓存策略主要分为**以数据为核心**和**以编码方式为核心**的两类，前者主要针对当前普遍存在的单速率编码的节目，后者则针对同一节目多速率编码的情况。以数据为核心的策略主要考虑的问题是在有限的代理缓存容量下，对哪些节目和节目的哪些部分进行缓存，以及如何对缓存内容进行更新替代，以使用户的平均起始延时最小和占用核心网络的带宽最少，并能支持用户的 VCR 类型的操作。由于占用核心网的带宽多少与需要从源服务器传送到代理服务器的数据量有关，因此也可以用字节命中率来描述这一指标。所谓**字节命中率**是指节目缓存的字节数与节目的总字节数之比。字节命中率越高，需要从源服务器索取的数据量越少。

通常我们对于用户对节目的点击率有一个经验的估计，即 20％的节目有 80％的点击率，而其余 80％的节目只有 20％的点击率，前者称为热点节目。热点节目一般都完整地存储在代理服务器中，非热点节目则只有每个节目的前一部分存储在缓存器中。当用户点播非热点节目时，代理服务

器向用户传输节目的前一部分，同时向源服务器索取节目的后一部分，以保证向用户提供持续的节目流。代理服务器在索取后一部分节目为当前用户提供服务的同时，还可以将此部分数据保存下来以供后续的用户点播，也可以将此部分用完后删去。

热点节目和非热点节目的前一部分通常由源服务器根据节目管理策略主动（用户发出请求前）发送给各代理服务器，这称为**“推”模式**；而用户点播非热点节目时，代理服务器向源服务发出请求，源服务器才向代理服务器发送数据，这称为**“拉”模式。**

在代理缓存中部分地存储一个节目的方法主要有两种。一种称为**前缀缓存**，即将节目分成前缀和后缀两段，只缓存前缀部分，此时前缀的长度至少要在代理服务器向源服务器发出请求到获得数据这段时间内，足够保证用户的正常播放。另一种方法称为**分段缓存**，即将节目分成长度不等的多个段，距离节目起始点越远的段长度越长。若第 0 段的长度为一个单元，第 i 段的长度则为 2^{i-1}；对于任何大于 0 的 i 值，有 2^{i-1}，$2^{i-1}+1$，…，$2^{i}-1$。与前缀缓存相比，在这种方法中节目被分成了更多的段，每段的长度减小，这使得在设计缓存替换策略时可以对较小粒度的数据块进行操作，因而具有较大的灵活性，有助于字节命中率的提高。但是这两种分段方式都不支持用户的 VCR 操作，更多的分段方式读者可以在相关文献中找到[9]。

随着用户的点播不断有数据从源服务器移进代理服务器；而随着时间的推移，热点节目可能变成非热点节目，那么某段时间内在有限的代理缓存器中应该保留哪些节目或哪些节目的哪些部分，则是**缓存替换策略**需要解决的问题。确定缓存替换策略的准则是提高该时间段内的平均字节命中率和降低平均用户等待时间；而决策通常以用户对某个节目点击的频率为依据，点击频率高的节目获得较大的存储空间，点击率很低的节目被删除。进一步的研究表明，在决定替换策略时，除点击频率之外，节目的播放速率也是需要考虑的因素[9]。

以编码方式为核心的缓存策略主要考虑分层编码（见 10.2 节）的节目的缓存方法。例如，将能够给出低质量图像的基本层数据放置在代理缓存器中，具有低质量终端的用户点播节目时，代理服务器直接从缓存区中输出数据；当具有高质量终端的用户点播同一节目时，代理服务器向源服务器索取增强图像质量的增强层数据，将增强层数据和本地存储的基本层数据一并传输给用户。

3. 合作缓存机制与内容路由

合作缓存是指各个 CDN 服务器上分别存储着不同的节目，当本地 CDN 服务器上没有用户点播的节目时，它可以通过内容路由从周围缓存了该节目的 CDN 服务器索取节目，这一方面避免了一个 CDN 服务器复制所有节目耗费过多的存储资源，也避免了所有服务器都向源服务器索取节目而使源服务器过负荷。图 8-52 给出了一个两层合作缓存的结构。地域相邻的服务器构成一个簇，簇内服务器相互连接，每个簇中有一个代表服务器，例如图中簇 A 中的 R_A，各簇的代表服务器互联构成第 2 层网络。

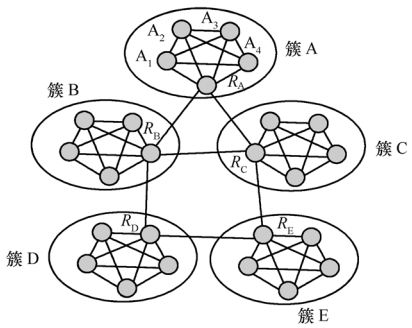


图 8-52　由 CDN 服务器构成的两层叠加网络

如果本地服务器上有用户点播的节目，我们称为本地命中；同理，如果本簇中的服务器和网上任何一个服务器能够提供用户点播的节目则分别称为簇命中和全局命中。总的本地命中数与 CDN 网总的用户请求数之比称为**本地命中率**，同样可定义簇命中率和全局命中率。这些构成了衡量合作缓存机制性能的重要指标。高的本地命中率和簇命中率表明可以节约更多的网络资源，减轻源服务器负荷，并且由于提供服务的服务器更接近于用户，网络可提供的 QoS 更高。

如果节目在各个 CDN 服务器上的分布是预先分配

好不变的，那么本地服务器在没有命中的情况下很容易找到具有用户所需节目的服务器，此时的内容路由是很简单的。如果节目在各服务器上的分布是动态变化的，那么内容路由就比较复杂。下面我们以簇内的内容路由为例介绍几种常见的方法。

(1) 基于查询的机制

如果本地服务器不含有用户请求的节目，那么它向簇内所有其他服务器发送查询消息，以找到具有该节目的服务器。这种方法的缺点是，网上需要传输大量的查询消息，而且只有在所有响应消息返回之后，才能最终确定簇内不存在该节目，有较长的延时。

(2) 基于摘要(digest)的机制

每个服务器保存一份节目摘要，该摘要中包括簇内其他服务器所保存的节目的清单。当某一个服务器增加或删除节目时，它告知其他服务器对各自的节目摘要作相应的修改，因此本地服务器根据自己的节目摘要就可找到具有用户请求的节目的服务器。这种方法的缺点是服务器需要频繁地相互交换摘要更新信息，否则 不能准确地了解节目的存储位置。

(3) 基于目录的方法

在这种方法中需要有一个目录服务器，目录服务器负责维护簇内所有服务器的节目信息。CDN 服务器在更新自己缓存的节目时只需要通知目录服务器，而在本地没有命中时也只需要向目录服务器发送查询信息。此时目录服务器成为一个关键部件，一旦损坏则整个系统将不能工作。

(4) 基于哈希表的方法

一个更有效的方法是簇内每个服务器都保存一个相同的哈希函数，根据节目的 URL(或其他唯一性标识)、服务器的唯一性标识(如 IP 地址)和哈希函数决定节目在哪个服务器上存储。在查找某个节目时根据哈希函数就可以很快找到存储该节目的服务器。但是纯哈希方法，即节目只存储在哈希函数决定的服务器上，虽然使得网络中没有任何冗余的拷贝，节目共享的效率很高，但是可能使本地命中率降低。因此哈希方法往往和其他方法混合使用。

当在簇内找不到用户请求的节目时，就需要进行簇间的路由。此时由于各簇的代表服务器在地域上相互较远，基于哈希表的方法不太适用；而基于摘要的方法要由代理服务器维护包括其他簇的巨大的目录也是十分困难的。

4. 基于代理服务器的应用层多播

在直播应用中，代理服务器作为多播服务的节点构成叠加在 IP 网之上的多播网络。我们知道在 IP 层用 D 类地址来标识多播组，而在应用层多播中用 URL 或其他与应用有关的关键项来标识多播组。所有多播组成员在叠加网络上构成**多播树**，应用层多播的关键问题就是如何构建多播树。

在用于直播的应用层多播中，每个节点(CDN 服务器)是单向传输的，一个多播组构成一个从源服务器出发的单源生成树。如果以延时作为性能的衡量指标，我们希望构建一个最小延时路径生成树，使源到树上每个节点的延时最小。由于端到端的延时通常由传播延时、传输延时和排队延时构成，当端到端的带宽得到保证时，传输延时和排队延时也就得到了限制，因此也可以以可利用的带宽作为性能的衡量指标，构建一个最宽路径生成树，使树链路的最小可利用带宽最大。考虑到核心网的带宽一般比较宽裕，CDN 服务器的接入带宽是限制它同时支持的多播用户数量的主要因素，因此许多算法都以接入带宽，或者接入带宽和延时作为构建树的衡量指标。

从路由的策略来说，应用层多播树可以是静态的，也可以是动态的。所谓静态即是根据对网络状态的长时间的估计，建立 CDN 服务器之间的固定的多播树关系，每当一个多播组形成就利用复盖该组成员的子树来传递节目信息。所谓动态即指每当一个多播组形成，就根据当前网络状态(如可利用带宽)构建多播树。显然动态方式能够更有效地利用网络资源，但在通信和控制上花费的代价要比静态方式高。此外，多播树的建立可以是集中式的，或者是分布式的。在集中方式中，由中央管理服务器负责收集网络状态信息(可利用带宽、延时等)和所有 CDN 服务

器的信息(CPU 能力、剩余接入带宽等),并在此基础上为每一个多播组建立多播树。在分布式方式中,每个 CDN 服务器收集与它相连的本地链路信息和它邻近服务器的信息,每个多播组构建自己的多播树。集中方式比较适合于小规模的 CDN 网络,而分布方式有更好的扩展性。在一次直播过程中,CDN 服务器可能随时加入或退出多播组。如果在每次发生这种情况时就重新构建一个多播树,称为重构方法。重构过程可能影响到仍在传输节目的服务器,因此需要采取适当的方法使仍在工作的服务器从原有树结构平稳地过渡到新的结构。另一种处理节点动态加入和退出的方法称为累积方法。在这种方法中,当一个 CDN 服务器加入多播组时,多播树增加一条单向路径将该服务器连进来;当一个服务器想退出时,如果它没有子节点,则可以从树中将自己删除;否则,它必须保留在树中,直至它的所有子节点都退出为止。可以看出在这种方法中,仍在传输节目的服务器不会受到加入和退出节点的打扰。

8.6.7 3GPP 分组交换流服务(PSS)

由 3GPP TS26.234 定义的**分组交换流服务 PSS**(Packet-Switched Streaming Service)是移动通信中进行流服务的标准,它的一个基本特点是能够根据网络条件的变化提供自适应的流服务。PSS 的应用场景与图 8-39 所示类似,它借鉴了有线网络流服务所使用的方法和协议(如 RTP/RTCP、RTSP 和 SDP 等),并针对无线网络的特点做了相应的扩展。

相比于有线网络来说,无线链路的带宽可能随时间有很快的变化,衰落或物体遮挡可能使信号瞬间衰减到接近于零。同时,不同的无线接入技术,如 WCDMA 或 CDMA2000,具有不同的最高平均可利用速率,因而在不同系统的小区间切换时,链路的特性可能有显著的变化。图 8-53 给出了在无线环境下 PSS 进行自适应流服务会话过程的示意图。它与图 8-39 所示的有线网的情况相似,其不同之处在于:(1)会话描述文件的传送方式不同;(2)明确定义了随网络状态变化进行自适应流服务的方法。

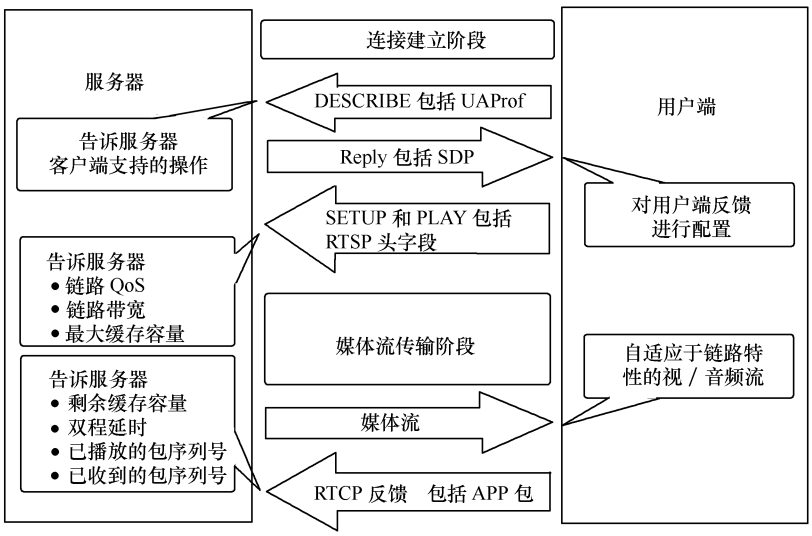


图 8-53 自适应流会话过程

用户首先通过向服务器发送 RTSP DESCRIBE 命令发起会话,该命令中包括 UAProf。**UAProf**(User Agent Profile)是**无线应用协议**(Wireless Application Protocol,WAP)的一个规范[10],它给出描述用户能力的词汇集,PSS 对该词汇集做了扩展。RTSP DESCRIBE 命令中的 UAProf 部分告诉服务器,申请会话的用户端能够支持自适应流服务的哪些操作(终端具备的能力)。服务器的 DESCRIBE 响应消息中包含一个 SDP 部分,该部分除了包括对将要传送的媒体流的描述外,

还包括与自适应流服务有关的用户配置信息，例如设置用户端缓存器充满程度的反馈及多长时间反馈、发送者(PSS 服务)和接收者 RTCP 报告可以占用的带宽等。在用户端进行了相关配置后，则可向服务器发送带有头字段的 RTSP SETUP 及 PLAY 等命令。3GPP 对 RTSP 的头字段的类型做了扩展。利用 Bandwidth 头和新定义的 3GPP_Link_Char 及 3GPP_Adaptation 头，用户端可以向服务器报告链路层 QoS 参数、期望的初始流传输速率以及自己的最大可利用缓存器空间等。服务器响应 SETUP 和 PALY 命令开始媒体流的传输，同时用户端定期向服务器发送 RTCP 报告，其中包括常规的 RTCP RR 和 PSS 定义的 NADU APP(Next Application Data Unit Application Packet)。RTCP 报告的主要内容包括用户缓存器剩余空间、包丢失率、已收到的包序号和已播放的包序号等。根据 RTCP 报告中已收到的包的最高序列号(HRSN)、下一个要播放的包的序列号(NSN)、下一个要播放的应用数据单元(音频帧或视频条)号(NUN)及缓存器剩余空间等参数，服务器可以了解接收端缓存器的当前状态(见图 8-54)。服务器根据接收端发来的 RTSP 和 RTCP 携带的信息来决定自己速率调整的策略。

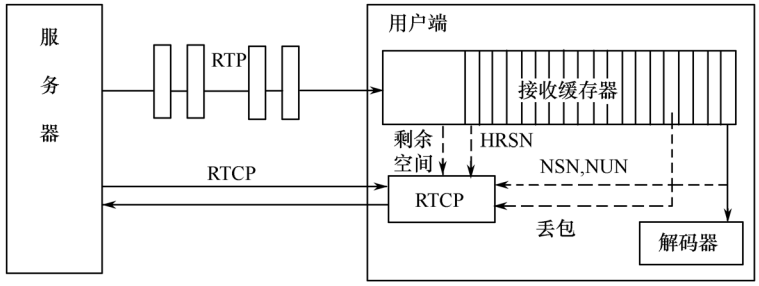


图 8-54　PSS RTCP 反馈信息及接收缓存器状态

服务器要进行速率调整，那么如 8.6.5 节所述，它存储的节目需要是多速率编码的或可伸缩性编码的。当链路可利用带宽下降时，服务器从传输高速码流改为传输低速率码流，以防止接收缓存器的下溢。3GPP 为在同一个文件中记录多个速率码流的情况规定了一个文件格式——**3GP**[11]。该格式不仅规定了记录不同速率码流的轨道和切换点，而且给出某个速率下推荐使用的轨道组合，以及应用数据单元与 RTP 包之间的对应关系，以使服务器在进行速率调整时可以直接对 RTP 包进行操作，而不必关心其中的媒体内容。

由于无线链路的状态变化较快，PSS 系统的接收缓存器要尽可能地保持在充满状态，以便在衰落、遮挡或小区切换造成数据传输短时中断时播放不至于中断。同样由于信道状态变化快，为了及时进行速率调整，PSS 规定 RTCP 报文的发送周期比有线网的情况要短，一般是一个双程延时(RTT)到 1 秒之间，服务器和用户端占用的 RTCP 带宽则分别限制在 4 kb/s 和5 kb/s。此外，PSS 还规定了相应的编码标准，其中 AMR 和 H.263 分别为强制性的音、视频编码标准，扩展的 AMR-WB(AMR－WB＋)、增强的 AAC＋和 H.264 分别为推荐的音、视频标准，其他标准还包括宽带语音 AMR－WB 和 MPEG4，以及 JPEG、GIF 图像标准、SVG 矢量图形标准和 SP-MIDI 电子音乐标准等。最后需要指出，PSS 是针对点播的场景设计的，对于不存在反馈信道的无线广播场景的流传输，可以参见 **3GPP MBMS**(Multimedia Broadcast/Multimedia Service)规范[12]。

8.6.8　IPTV

IPTV 通俗地讲就是通过 IP 网看电视；按照国际电联给予的比较严谨的定义则是：在达到所要求的 QoS、安全、交互性和可靠性水平的可管理的 IP 网上传递包括电视、视频、音频、图形和数据等的多媒体业务。由于 IPTV 系统也是一种要求视/音频数据边传输、边播放的流式应用系统，因此所涉及的基本技术与前面介绍的一般流媒体系统相同。它与通常所说的流媒体的区别是，流媒

体通过一般因特网传输、一般在微机或手持设备上观看；而 IPTV 是在可管可控（具有一定 QoS 保障）的 IP 网上传输、具有广播电视质量（可在电视机上观看）的流式应用系统。为了保证网络的可管可控，IPIV 通常只部属在同一家网络提供商的网络上。

1. IPTV 系统的构成和基本业务

IPTV 既然是电视业务就应当像传统的广播电视那样提供较高的图像质量，并且具有从内容提供商到用户家庭的传递和分发节目的完整机制。在一般流媒体系统中，一条流的速率在几十 kb/s～几百 kb/s，而 IPTV 的带宽则应高于 1 Mb/s，对于 HDTV 可能在十几 Mb/s～几十 Mb/s。为了保证较高的图像质量，对网络的时延和丢包率也有较高的要求。图 8-55 给出了一个 IPTV 系统的基本组成部分，其中内容提供商制作并拥有具有自主版权的节目；服务提供商从内容提供商处获得节目，并将其纳入自己的业务提供给用户；网络提供商提供从服务提供商到用户之间的网络连接；用户则消费和享受节目。在用户端，可能只有一台终端设备，也可能是连接多台设备的家庭网络。

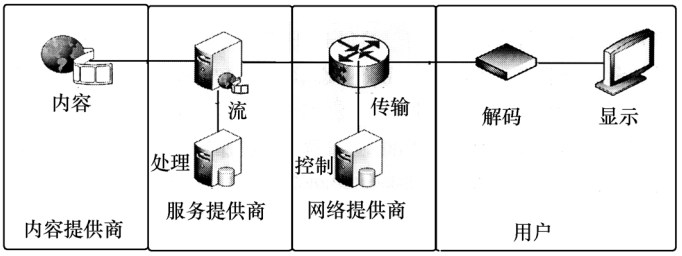


图 8-55　IPTV 系统参考模型

IPTV 的基本业务包括：(1)直播电视。服务提供商通过传统的广播传输途径（如卫星、电缆或地面广播）接收内容提供商提供的节目，并将其转换成自己的业务所规定的格式后，以多播形式通过网络提供商提供的传输网络和用户家庭网络将节目传送到用户端设备上；(2)点播电视。服务提供商通常通过物理介质（如磁带、光盘等）从内容提供商处获得节目，将其转换成自己业务所规定的格式后存储在本地服务器内，然后根据用户的需要以单播方式通过网络提供商提供的传输网络和用户家庭网络将节目传送到用户终端设备上；点播电视允许用户进行录像机式（快进、快退、暂停等）操作；(3)混合模式。用户通过传统广播电视传输通道直接从内容提供商接收电视节目，同时通过网络提供商提供的传输网络获取点播内容的服务。这种形式允许在收看电视广播的同时，获得相关的辅助信息或进行交互操作。

2. IPTV 的基本功能模块

图 8-56 给出了一个 IPTV 系统的基本功能方框图。它包括头端、传输网络和用户终端三个部分，其中头端由服务提供商运营，传输网络包括网络提供商提供的传输网络和用户家庭网络。头端通过某种方式从内容提供商获得节目。头端中的直播电视服务器和 VOD 服务器分别负责向用户提供如前所述的直播和点播服务。头端的中间件与用户端的中间件是实现视频流传输的一组软件，它以服务器/客户端的结构工作，负责用户与头端的会话管理和应用管理之间的交互操作。会话管理借助于资源管理建立用户和 VOD 服务器之间的连接，资源管理为建立连接的用户预留资源。应用管理提供交互机制使用户可以和头端直播电视服务器建立连接；应用管理还负责提供一些特定的服务，如交互游戏、账单显示等。头端中间件的其他功能还包括数字版权保护、加密和**电子节目表 EPG**(Electronic Program Guide)生成等。头端中的用户管理负责维护用户和收费信息。网络提供商提供的传输网络可以分为核心网和接入网。在核心网上实际上需要构架一个内容分发网络，以将各个“频道”的节目传送到网络边缘的 2 级头端。IPTV 的用户接入网主要为 7.10 节中

介绍的各种形式的 ADSL 和 VDSL 网络。在核心网和接入网接口处，为了提供 IPTV 业务还需要实现基于 IP 的**条件接入(CA)**，并支持视频流传输的多播与 QoS 功能。用户端设备通常是机顶盒加上电视机，也可以是计算机。机顶盒完成连接建立、频道切换、解码和各种控制功能，然后将重建的图像信号传送给电视机显示。如果用户端采用家庭网络，则可以连接数个终端。

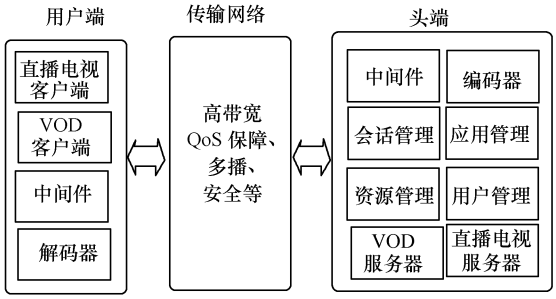


图 8-56　IPTV 功能模块

3. 因特网流媒体联盟的 IPTV 体系结构

因特网流媒体联盟 **ISMA**(Internet Streaming Media Alliance)是一个由有关工业和企业组成的国际标准化联盟，致力于制定流媒体的相关标准。它已经制定的两个最重要的标准是 ISMA1.0 和 ISMA2.0，图 8-57 为 ISMA2.0 给出的体系结构。

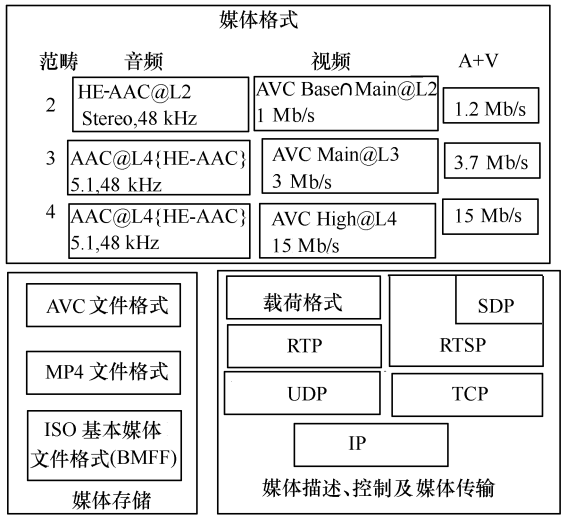


图 8-57　ISMA 2.0 体系结构

由图看出，ISMA 主要规定了媒体格式、媒体描述、控制、传输和存储方面的标准。在媒体格式方面，ISMA 2.0 规定了图示的三个范畴(profile)，分别对应于 H.264(AVC)的不同范畴的层次，它与 ISMA 1.0规定的 2 个速率较低的 MPEG4 的范畴并不兼容。在媒体控制和传输方面，ISMA 采用了 IETF 的 RTSP 和 RTP/UDP 协议；作为可选项，也允许 RTSP 和 RTP 交错在 TCP 上传输，此时传输是可靠的且易穿过防火墙等设施，但在实时性方面则不如广泛使用的 RTP/UDP 方式。在媒体描述方面，ISMA 采用 SDP 对媒体的元数据和播放起始延时等信息进行描述；SDP 描述信息可以通过多种形式，如 HTTP(网页)、RTSP、SMTP(邮件)、SAP 或 SIP 传递给用户。在媒体存储方面，ISMA 规定的文件格式为 ISO/IEC 14496—12 规定的 **ISO 基础媒体文件格式**(Base Media File Format，BMFF)及其两种实现：MP4 和 H.264 文件格式[13]。

在 ISMA 系统中，音、视频采用各自独立的 RTP 流传送，两者在 UDP/IP 端口上复用。采用

独立的流有利于没有视频接收能力的终端单独对音频流进行接收。服务器和用户端在发送和播放两个流时，需要根据 RTP 时间戳完成音、视频流的流间同步。

4. 欧洲电信标准协会的 IPTV 体系结构

欧洲电信标准协会 **ETSI**(European Telecommunications Standards Institute)于 2005 年提出了一个 IPTV 的方案(ETSI TS 102034)，它所涉及的协议栈如图 8-58 所示。由图看出，该方案的特点是将视、音频和用户数据先复用成一个 MPEG TS 流，然后再封装成 RTP 包在 UDP/IP 上传输(见图中部)。视、音频流之间的同步无须借助于 RTP 时间戳、而是在 TS 复用时完成。这种格式与欧洲数字视频广播 **DVB**(Digital Video Broadcasting)标准的流传输格式相同，因而也称为 DVB over IP。

业务发现和选择(Service Discovery and Selection，**SD&S**)是 ETSI 方案的另一个重要组成部分，它定义一套方法使用户能够寻找到有哪些业务可供选择以及如何选择自己所想要的业务。其具体的做法是，用户终端启动时先到一个固定的地址(称为入口点)寻找 SD&S 信息，该地址是一个众所周知的地址或者通过配置文件获得。SD&S 信息是由{服务提供商 DNS 域名、所提供的业务名称、该业务的节目列表、节目描述信息地址、节目内容地址}构成的一个树形结构，该结构每隔 30 秒循环一次。用户通过 SD&S 信息找到自己想看的节目后，如果节目是直播业务，用户可以利用适当的 IGMP 命令加入或退出节目多播组；如果节目是点播业务，则使用 RTSP 命令进行点播控制。ETSI 为 SD&S 在 IP 多播上的传输定义了一个轻量级的传输协议，称为 DVB STP(DVB SD&S Transport Protocol)；SD&S 在 IP 单播上的传输采用 HTTP 协议(见图 8-58 中左)。

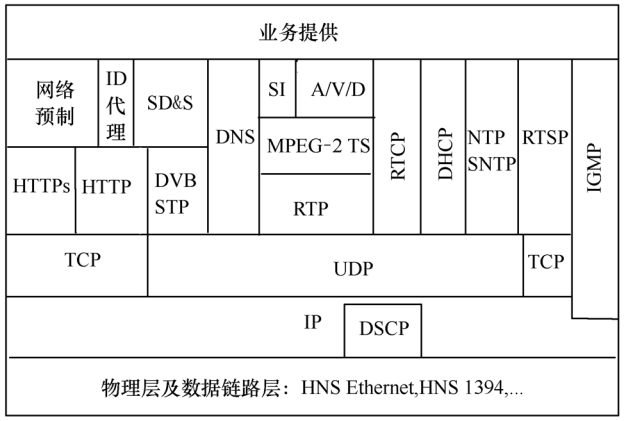


图 8-58　DVB IP 业务的协议栈

在 ETSI 方案中用户终端的 IP 地址是动态 IP 地址，由 DHCP(Dynamic Host Configuration Protocol)服务器管理。用户终端还需要通过 SNTP(Simple NTP)或 NTP 支持实时时钟。图 8-58 所涉及的其他协议还包括：(1)ID 代理。每一个用户终端有一个 ID，便于服务提供商对其进行网络配置，配置信息通过 HTTP over TLS(Transport Layer Security)传输；(2)服务信息 SI(service Information)。我们在 8.2.2 节中介绍过 TS 流中的 PSI，PSI 包含了对当前码流的描述。为了使用户对其他时段(当前未播)的节目有所了解，DVB 对 PSI 进行了扩展，在 PSI 的 4 个表之外又增加了业务群关联表 BAT、服务描述表 SDT、事件信息表 EIT、时间日期表 TDT 等 9 种表。所有这些表组成**服务信息 SI**，并打包复用进 TS 流。同时，利用 SI 中的 NIT、SDT、EIT 和 BAT 可以生成 DVB 的电子节目表 EPG 的基本部分。在 ETSI 方案中，如果 TS 流中包含了 SI，则 SD&S 部分可以简化。

5. 国际电联的 IPTV 体系结构

在国际电联有关 IPTV 体系结构的讨论中，将图 8-55 所示的参考模型细化为图 8-59 所示的结

构框架，其中内容提供商和用户终端保持不变，而服务提供商和网络提供商的功能被集成为三个层次：应用层、业务层和网络层。图左侧的用户端分为终端设备和用户传输设备两部分，终端设备（如机顶盒）负责节目选择、交互控制和媒体流接收与播放等；用户传输设备是用户端接入网终接点到终端之间的网络连接设备。图中部的应用层为用户提供了解、选择或购买节目的手段，并支持节目内容的保护；业务层包括媒体流的产生和相应的控制；网络层分为网络控制和网络传输两部分，前者提供 IPTV 所需的 QoS 和安全性，后者提供 IPTV 业务和用户传输设备之间的 IP 连接。图右侧除了内容提供商之外，还有操作支持系统 OSS（Operational Support Systems）和业务支持系统 BSS（Business Support Systems），二者负责系统的成功运作和管理。图中的各方框表示出具体的功能模块。

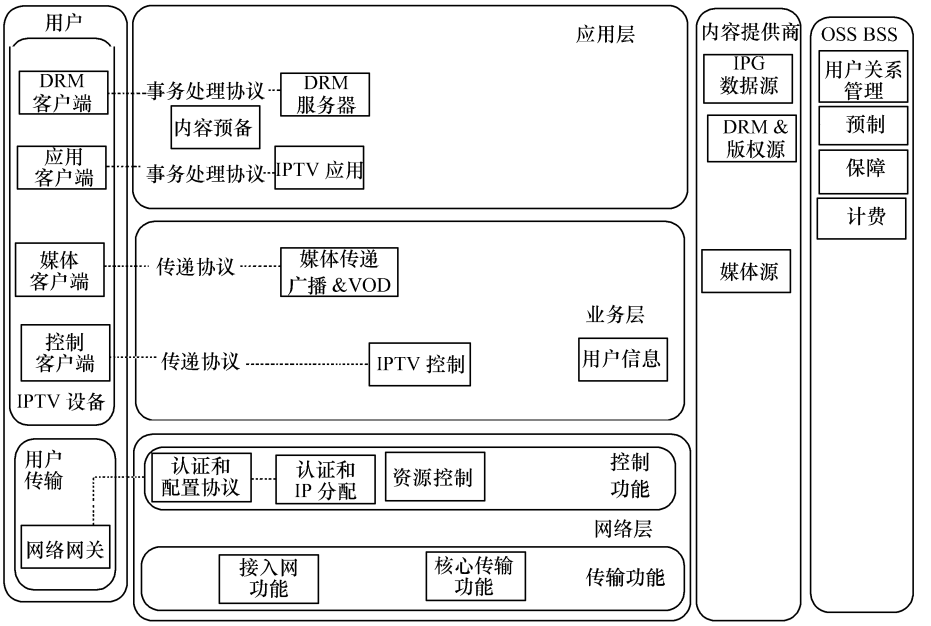


图 8-59　国际电联 IPTV 结构框架

在上述框架的基础上，ITU-T 定义了如下的 3 种 IPTV 系统。

（1）基于 NGN 结构和 IMS 会话控制的系统

这种系统建立在图 7-45 所示的 NGN 体系结构之内，并利用图中 IMS 业务部分进行会话控制。此时，图 8-59 中的网络层的控制和传输部分分别对应于图 7-45 中的 NACF、RACF 和接入、边界、核心网络；图 8-59 中的业务层则分为业务支持层和业务控制层，业务支持层负责流的传输，业务控制层利用 IMS 核心（SIP）实现与用户终端设备之间的会话建立和控制。这种方案相当于利用图 7-45 中的 IMS 业务部分实现 IPTV 业务。

（2）基于 NGN 结构和非 IMS 会话控制的系统

这种系统除了会话控制之外与上面的系统相同，它相当于利用图 7-45 中的流媒体业务部分实现 IPTV 业务。它不通过 IMS、而是有自己建立会话的机制，例如直接将内容列表和服务器 URL 传给用户。

（3）非 NGN 系统

这种系统不具有图 7-45 所示的体系结构，它在所有层次上都有自己定义的机制。

从上面的例子可以看出，ISMA、ETSI 和 ITU-T 对 IPTV 的标准化有着不同的侧重面；还有其

他一些国际标准化组织，例如 ATIS IIF(IPTV Interoperability Forum)、DSL Forum 和我国的中国通信标准化协会等，也在制定覆盖面各异的相应标准。不过根据前面的介绍我们可以知道，IPTV 的标准将涉及业务总框架、音/视频编码、数字版权保护(DRM)、内容分发、电子节目表、网络及终端技术等内容。

8.7 对等网络(P-to-P)流式应用系统

对等(Peer-To-Peer，简称 P2P)网络是相对于传统的客户机/服务器(C/S)结构而言的，在 P2P 网络中，用户设备既是终端，又是网络节点，相互之间是对等的。当一个用户想要某个节目时，它不是像在 C/S 结构的流式系统中那样直接从服务器获得数据，而是从曾经接收到过(或正在接收)相同节目的其他多个用户(**节点**)获取数据；在它收到这些数据后，也可以向后续的请求该节目的用户提供数据，即每一个节点既可以作为客户机，也可以作为服务器。这种做法极大地减轻了造价昂贵的流服务器的负担，可以共享各个节点的计算、存储和带宽资源，节约网络建设的费用，增加网络传输的可靠性，并能够很容易地进行系统规模的扩展，无须利用 CDN 就可以达到百万用户的规模。P2P 技术近年来已广泛应用于文件共享、即时通信和 VoIP 中，它在流媒体系统的实际应用也已经获得了巨大的成功。随着 P2P 技术的发展，未来很可能网上的许多业务都可以由 P2P 技术来支持。

8.7.1 工作原理

图 8-60 给出一个典型的 P2P 视频流式系统的示意图。系统包括：(1)一组视频源服务器。在直播时它转发从卫星、地面或电缆电视系统获得的视频流；在点播时提供存储的视频节目；(2)一组**登录服务器**。它是业务的入口，用户(节点)通过它获知节目表和选择节目，并得到与之距离最近的跟踪服务器的地址，及其他一些功能，如穿越防火墙等；(3)一组**跟踪服务器**。它们保存着所有节点及其持有数据的信息，当有节点查询时，向查询节点返回可供其下载数据的相邻节点的列表；(4)众多的可以自由加入和退出系统的节点，可能是微机、智能手机或其他智能设备。当某个节点获知自己可以下载数据的节点列表后，则与表中节点建立通信连接，并获得自己需要的数据。图中的箭头表示出一个新节点进入系统的建连过程。

为了让多个节点参与同一个节目的数据传输，P2P 系统通常给每一个节目分配一个标识 ID，然后将其分成段，称为**大块**(Chunk)。同一个节目的大块具有相同的 ID，并按播放的时间顺序标以序列号或时间戳。根据这些信息，节点可以知道自己已收到和未收到哪些数据段(大块)，并用一个称为**缓存图**(Buffer Map，简称 BM)的数据结构记录下来。

如图 8-61 所示，每一个节点具有两个(逻辑上的)缓存器，一个用于本地播放，一个用于参与 P2P 节点之间的数据传输；图中虚线框所示的部分我们称为 **P2P 流引擎**，负责与跟踪服务器和邻近节点的通信联系。节点通过自己的 BM 知道本地播放缺失哪些数据块，并尽力在播放时限之前，从跟踪服务器提供的可供下载这些缺失块的节点处获取数据，以保证播放的连续性。而跟踪服务器则需要各个节点周期性地向它报告自己的状态和 BM，以便向查询节点提供邻近节点的列表。在播放时限之后，数据块从播放缓存器中删除，但还可能继续保留在数据包缓存区中，以供其他节点下载。在直播应用中，所有节点观看的同一节目基本上处于相同时间段，节点之间较容易共享相同的数据，因而数据包缓存区容量较小。在点播的情况下，用户观看的段落可能分散在节目不同的时段上，此时为了节点之间有充分的数据可以共享，数据包缓存区的容量需要较大，可达 1GB(大部分在本机硬盘上)。

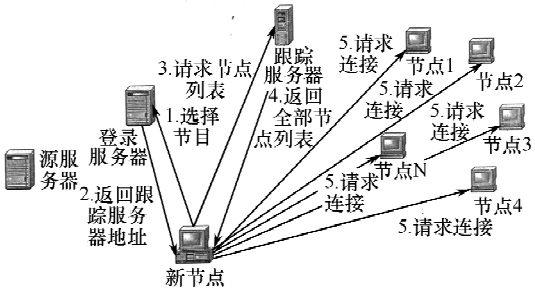


图 8-60　P2P 流系统示意图

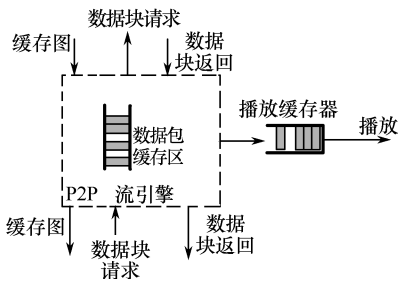


图 8-61　P2P 节点的流引擎

由上看出，P2P 系统需要完成如下三项基本功能：(1)**发现**。使新节点了解系统提供的业务(节目表或频道列表)、该项业务的元数据(文件标识、大小等)和可供下载数据的节点列表；(2)**走位**。使新节点了解跟踪服务器和可供下载数据的节点的位置(IP 地址及端口号)；同时新节点也必须报告自己的位置和已存储的数据。这些定位(地址与 BM)信息定期地交换，使跟踪服务器所保存的信息得以及时地更新；(3)**数据传输**。节点下载自己所需要的数据和上载其他节点需要的数据。不同的系统采用不同的数据传输方式和不同的网络拓扑结构。一种传输方式是**“推”**，即由进行上载的节点决定数据的接收者；而另一种方式为**“拉”**，即由进行下载的节点向一组可提供数据的节点提出数据下载的请求。在节点网络连接拓扑方面，则可以分为基于树结构和基于网结构(mesh)的两种，后者是一种自组织网络。

8.7.2　网络结构与数据传输

1. 基于树结构的方法

早期的 P2P 系统多采用树结构的网络。在这种网络中，对等节点像 CDN 应用层多播那样，以源服务器为根构建一个或多个多播树，媒体流以“推”的方式，从根逐级推向树的末梢。各个中间节点通过其下行信道从父节点接收数据，其上行信道则用来向子节点传输数据。这种方法保证了数据的单路由传输，有较高的传输效率。但是由于利用用户终端来作树的节点，因此存在两个问题：①众多用户随时进入或退出系统造成树结构的高度不稳定；②用户终端的接入信道一般是不对称的(如 ADSL)，上行信道较窄，如果下端节点只通过一个分支接入上游节点的话，则节点接收到的媒体流速率由它所在树枝上游节点的最小上行信道带宽所限制，从而影响它获得的视频图像的质量。

构建多个多播树可以在一定程度上解决上述问题。图 8-62 给出一个具有三个多播树的例子。本例中，假设了上、下行信道带宽相等。通常源服务器有足够的带宽同时支持多个媒体流的传输，源服务器可以同时传输的媒体流的个数决定了多播树最多的个数。源服务器将一段时间(一个时间窗口)内的数据分成几段，段数等于多播树的个数，在图 8-62 中为 3。源服务器将各段数据分别推送给每个多播树，对于下一个时间窗口的数据也作同样的处理。在图 8-62 中，R1、R2 和 R3 代表各段构成的数据源，实线、虚线和点画线分别代表各段数据的传输路径。以图中节点 7 为例，假设节点 4 退出系统，节点 7 仍然能够收到 3 段数据中的 2 段。显然分段数越多，也就是多播树个数越多(传输路径多)，因节点退出而引起的数据丢失量越少，数据传输的鲁棒性越强。

从理论上说，树这样结构化的网络，便于按一定准则对其性能进行优化，有较少的传输开销和下载延时。但是由于节点的自由出入，网络拓扑的动态变化使得它的实际性能并不理想和稳定。除此之外，处于树叶上的节点的资源也没有得到利用。

2. 基于网结构的方法

在这种网络中，节点周期性地以“拉”的方式向具有自己所需数据的一个或多个邻近节点(父节点)索要数据，这就构成了一个有向网(directed mesh)，其有向边表示了相应的父子关系。我们以

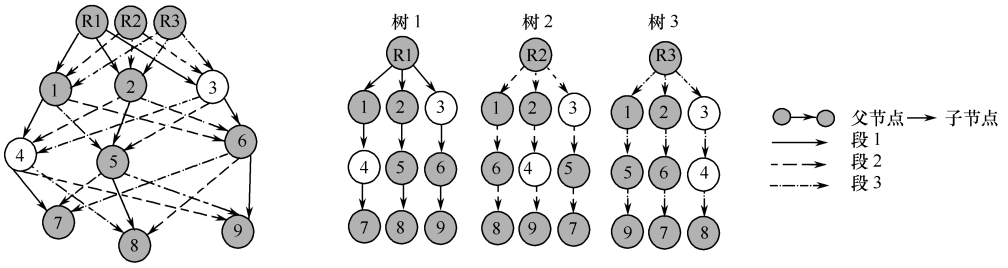


图 8-62　基于树结构的 P2P 流传输

图 8-63 为例说明流传输的过程。当一个节点需要数据时，它通过某种通信协议获得一个邻近节点地址的列表及每个邻近节点的 BM 的列表。如图 8-63(a)中，节点 5 为索要数据的子节点，1、2、3 和 4 为它可能的父节点，节点旁边的大括弧表示在当前时间窗口内它们各自的 BM。节点 5 获得邻近节点和它们的 BM 的列表后，开始数据索取的调度，调度必须满足它的播放时限。图中给出了节点 5 的调度表，如向节点 1 请求数据块 1，向节点 3 请求数据块 2，等等。在完成这一个时间窗口的数据索取之后，父子节点交换 BM 信息，如果父节点离开网络或不具有子节点需要的数据，子节点寻找新的父节点，然后开始下一个窗口的操作。由上看出，每个节点独立地选择自己的一组父节点，其父节点应具有它需要的数据段，并有空余(未分配给其他子节点)的上行信道带宽。同时我们看到，有向网的拓扑是周期性地更新的，与基于树结构的方法相比能更好地适应网络条件的变化，但为此付出的代价则是：①需要频繁地交换定位信息；②加大了数据传输的延时，因为整个过程包括由父节点向子节点传输 BM、子节点向父节点请求某个数据段以及最后实际的数据传输。

如何获得父节点列表来构成有向网，一般有两种方法，一是完全随机和分布式的，另一种是在中央服务器的协助下自治形成。在第一种方法中，节点通过泛洪机制来发现和定位其他节点，即节点首先向其相邻节点发出查询消息，如果某个相邻节点拥有它需要的资源，则返回响应消息；否则，将查询消息转发至自己的相邻节点；如此进行下去，直至消息的生命期(TTL)过期。由此可见，通信的开销是很大的。有的系统通过引入“超级节点”来降低这种开销。不过，许多著名的 P2P 视频流系统，如 PPLive、Joost、PPstream 等，都采用了第二种方法，也就是 8.7.1 节中介绍的典型系统，其中协助组网的中央服务器就是跟踪服务器。通常跟踪服务器在所有观看同一节目的节点中选择一个子集，构成可能的父节点列表返回给查询节点，该节点从列表中选择一个子集发出建连请求。在父子节点的通信连接建立之后，两者交换 BM，然后子节点进行数据索取的调度和数据传输。为了网络管理的需要，每个节点也定期地向跟踪服务器报告自己的状态、BM 及其他统计信息。

一个节点的父节点越多，传输路径越多，与树结构方法中多播树个数越多的情况一样，数据传输的鲁棒性越强。此外，选择什么样的父节点和数据传输的调度也是值得考虑的问题。我们以图 8-63(b)中两个数据段的传输为例加以说明。图中 R1 和 R2 代表源服务器直接支持的两个节点，结构 1 和 2 表示了两种不同的父节点的选择方法。在第 1 种结构中父节点向两个子节点传输不同的数据段，而第 2 种结构中父节点向两个子节点传输相同的数据段。从节点 5 接收的数据不难看出，第 2 种结构具有较小的深度，即较短的延时。由此我们可以得出结论，父节点向较多的子节点传送较少部分的数据有利于减少传输路径的长度，因此具有高上行带宽的节点应该放置在与源接近的位置上。不过在现有的实际系统中，跟踪服务器返回的父节点列表和子节点建连的父节点都是随机选择的。

8.7.3　其他设计原则

1. 数据块尺寸的选择

从数据传输的调度出发，希望将节目划分成尽可能小的数据块，以增加调度的灵活性，特别是

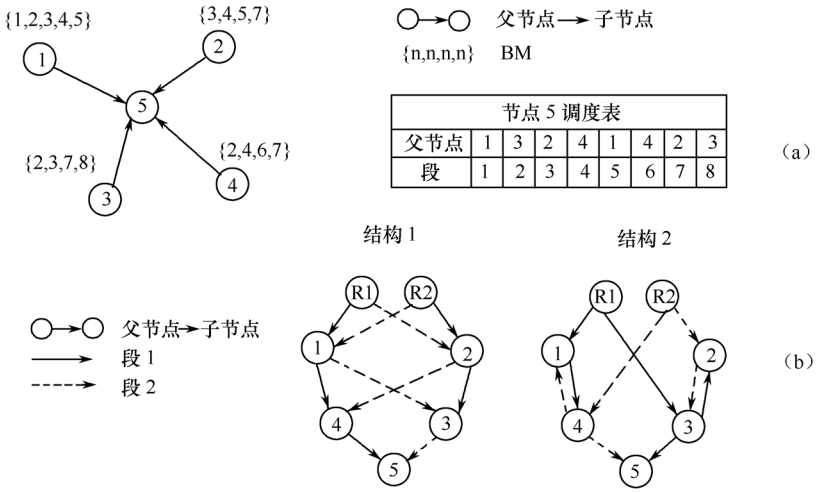


图 8-63　基于网结构的 P2P 流传输(a)“拉”数据调度;(b)不同父节点的选择

在节点的上行信道带宽不同的环境下,有利于提高上行信道的利用率。从降低开销的角度而言,希望数据块的尺寸越大越好。这些开销包括:①数据块的头信息(含有序列号、时间戳和认证信息等);②BM 的频繁交换;③与父节点建连的通信。块越大,一个节目分割的块数越少,表示 BM 所需的比特数越少(如分成 4 个块,BM 为一个 4bit 的矢量),需要连接的父节点数也越少,因此上述 3 项开销均小。而从保证流媒体的实时性出发,则希望把一定大小的块在其播放时限之前送进播放缓存器,这个块应该大到可以观看,而太大又可能致使传输层尚未收集齐块尾部的数据时,块前部的数据已超过了播放时限。

为了满足以上三方面相互矛盾的要求,一个节目通常首先被分成**大块**,大块用来构建 BM。如果大块大小为 2MB,一个 2GB 文件的 BM 为 1kb。大块进一步划分成**块**,块作为媒体播放器的处理单元。假设块大小为 16kB,对于 380～500kb/s 的视频流,1 个块相当于 1 帧左右的数据。块又分成**子块**(例如 1kB),子块是数据传输与调度的最小单元,以期有较高的传输效率和较小的丢包率。

2. 复制策略

众多节点的数据包缓存区共同构成了一个分布式的 P2P 存储(文件)系统,大块是硬盘存储的基本单元。节点只有拥有一个完整(数据齐全)的大块,才能通过 BM 向其他节点发布和传输该大块的数据。复制策略的目标是以尽可能小的代价,让分布式 P2P 存储系统上的数据能够为尽可能多的节点服务。

首先我们考虑节点缓存的节目数量。单节目缓存意味着节点只缓存正在观看的节目;多节目缓存则是除缓存观看的节目外,还可同时缓存并向其他节点上载另一个节目。其次要考虑的是,是否允许数据的预先提取。如果不允许,节点数据包缓存区中就只有该节点已经观看过的数据。显然,在多节目缓存中,必须允许数据的预取,但同时下载另一个节目可能会影响正在观看的节目。最后要考虑的是,当节点的数据包缓存区充满时,应该删去缓存区中的哪一部分数据。可能采取的策略有删去最早使用过的节目,或最少使用的节目。通常数据的删除以节目为单位,而不是以大块为单位进行,因为后者需要记录有关大块的信息,开销比较大。

3. 下载数据块的选择

在给定 BM 的条件下,节点从相邻节点优先下载哪些数据块,可以有如下三种选择:①播放时限最近的大块;②系统中最稀缺的大块;③能够独立解码(如包含Ⅰ帧)的大块(在 VOD 需要快退/快进操作时)。在通常的作法中,第一种选择有最高的优先级,第二种其次。后者可以加速稀缺的

扩散，间接地提升系统的性能。

4. 传输策略

由于相邻节点的状态是动态变化的，节点在获取缺失数据块时必须考虑最大化下载速率和最小化开销。可能的做法有：①从多个邻节点同时下载同一个数据块；②从多个邻节点同时下载多个不同的数据块；③从一个邻节点下载所有所需的数据块，只在该节点不可用时才转向另一个邻节点。显然，第一种策略有利于保证播放时限的满足，但产生重复传输的可能性远高于后两种策略。当所有邻节点都没有自己需要的数据块时，节点可以与源服务器建连直接获取所需数据。

5. 其他

一个完整的 P2P 视频流系统还需要考虑网络拓扑的管理与控制、穿越防火墙与 NAT(Network Address Translation)网络、内容认证与加密，以及激励等问题。所谓**激励**，就是使用某种机制让节点在花费资源参与 P2P 合作后，能够及时地获得一些好处，这将激励节点的合作性。目前在一些 P2P 文件共享系统中已经存在激励机制，而商用的 P2P 视频流系统中尚未有使用。

8.8 动态自适应 HTTP 流式应用系统

8.8.1 HTTP 流传输技术的发展

近年来因特网上的视频应用迅猛发展，基于 RTP/RTSP 的流式系统在一些方面上难以满足应用的需求。RTP 流要求服务器为每一个客户端维护一个独立的会话，这使得流服务器的造价很高，不利于大规模系统的部署和扩展。另外，在许多情况下，RTP 流不能穿越防火墙和 NAT 网络。因此随着因特网带宽的不断增加，人们更倾向于使用协议成熟、布设广泛、价格低廉的 HTTP 来进行媒体流的传输。但从 8.6.2 节对顺序式流传输的介绍我们知道，由于因特网的“尽力而为”特征，用户可利用的带宽很不稳定，因而 HTTP/TCP 流只适合短视频的传输。在传输长视频流时，需要有较长的(几秒到几十秒)起始延时来预存数据，并经常发生播放的停顿，以便积存数据支持后面的连续播放。

如果将长视频文件分割成像短视频那样长的段(远大于 RTP 包)，分段传输；同时在整个流的传输过程中，通过选择编码速率不同的分段来动态地调整媒体流的速率，以适应当前的网络带宽，应该是实现实用的 HTTP 流传输的一个可行方案。基于这种思想，一些公司近年推出了相应的系统，例如 Apple 的 HTTP Live Streaming、微软的 Smooth Streamig 和 Adobe 的 HTTP Dynamic Streaming 等。但是这些系统都是封闭的，有各自不同的文件分割方式、解析方法和客户端协议等。为了推动 HTTP 流传输技术的广泛应用，MPEG 与 3GPP 和 OIPF(Open IPTV Forum)等组织密切合作，于 2011 年底公布了 ISO/IEC 23009—1 标准，也称为 **MPEG—DASH**(Dynamic Adaptive Streaming Over HTTP)标准。

DASH 标准规定了媒体演示的描述信息(即元数据)的格式，称为 **MPD**(Media Presen tation Description)，以及媒体文件分割的数据段的格式。这使得符合标准的不同厂家的客户端和服务器之间可以进行媒体流的传输。图 8-64 给出 DASH 的应用场景的示意。图中，在广域网上利用现有的 HTTP 服务器和 HTTP 代理服务器进行媒体内容的分发；不同的终端，从智能手机到高清电视机，通过无线或有线接入网接入，都可以经因特网获得媒体流数据。

8.8.2 系统的工作原理

1. 自适应流切换

在 DASH 中，服务器上保存有一组或多组可供切换的媒体流，我们称为**自适应集**(Adaptation

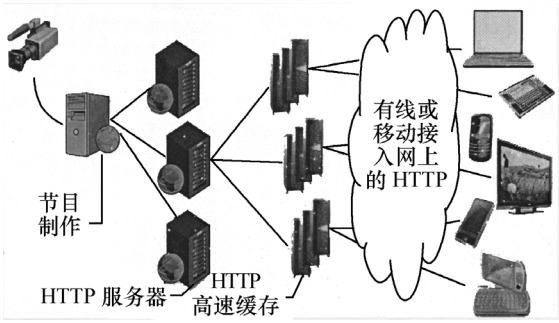


图 8-64　DASH 的应用

Set)。例如在图 8-65 所示的例子中,有视频和音频两个自适应集。视频集中含有同一节目的 5Mb/s、2Mb/s 和 500kb/s 编码的码流,以及一个**特技模式**(Trick mode)码流,它只含帧率很低的I帧,专供快进、快退等 VCR 操作时使用。音频集中含有同一节目的环绕立体声、128kb/s 和 48kb/s AAC 编码的 3 个中文伴音和 128kb/s 和 48kb/s AAC 编码的 2 个英文伴音。

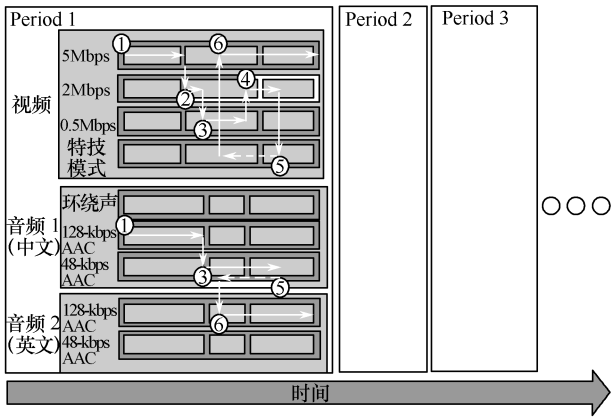


图 8-65　动态自适应速率调整的例子

假设用户终端不支持环绕立体声,在节目开始时索取的是最高速率的视频(5Mb/s)和 128kb/s 的中文伴音。当监测到网络带宽下降后,终端改为索取 2Mb/s 的视频段(见图中②);随着网络带宽下降到不足 2Mb/s,终端再次切换到 500kb/s 视频和 48kb/s 的伴音(见③);而后监视到网络带宽回升,切换回 2Mb/s 的视频(见④)。播放一段时间后,用户进行暂停和快退操作,此时终端开始索要特技模式的视频,而伴音则是静默的(见⑤);然后用户继续播放英文伴音的 5Mb/s 视频(见⑥)。

利用自适应流切换还可以实现多视角视频和可伸缩编码视频(见 10.2 节)的切换、3D 视频、不同字幕的叠加、广告动态插入和实时转播频道切换等。

2. 系统结构

图 8-66 是 DASH 系统的基本结构。服务器上的每一个媒体流按时间顺序划分成若干个周期(period),每个周期包含有一个或多个数据段(segment),每个段都有自己的统一资源定位符 URL。在每个周期内可能有一个或多个自适应码流集供客户端选择。一个节目的 MPD 给出节目及其周期和段的描述数据,以及包含的自适应集等信息。DASH 客户端首先通过 HTTP GET 请求/应答消息从服务器获得节目的 MPD;客户端也可以通过电子邮件等其他方式(标准并未规定)获得 MPD。为了提供更多的灵活性,MPD 还可以分成若干部分,通过多次下载获得。客户端获得并解析 MPD 后,按照自己的终端能力、当前网络状况和个人的喜好发出请求索取想要的数据段。

由于每个段都有自己的 URL，因此可以通过 HTTP GET 或带有字节范围的 Partial GET 请求/应答消息获得任何一个想要的段。客户端将收到的段组合成流进行播放，并继续根据监测到的当前网络带宽，按时请求自适应集中合适的后续数据段。由此看出，与 RTP/RTSP 流式系统中服务器以“推”模式进行流传输的方式不同，在 DASH 中，HTTP 是一个无状态的协议，会话由客户端管理。客户端以“拉”的方式从服务器获得数据，这极大地减轻了服务器的负担。同时，流速率的调整无须与服务器协商，使播放和动态速率调整很及时。

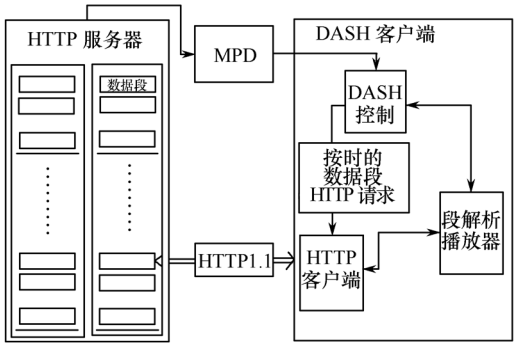


图 8-66　DASH 系统结构

DASH 可以利用现有的标准 HTTP 服务器、代理服务器及 CDN 进行工作。几乎所有的防火墙和 NAT 网络支持 HTTP 连接。如前所述，DASH 也能够像 RTP/RTSP 系统一样支持快进、快退等 VCR 式的操作。此外，现有的顺序式流传输系统很容易升级到 DASH。因此可以预见，在 MPEG 对 DASH 制定标准之后，基于 HTTP 的流媒体系统会在近几年有一个蓬勃的发展，并成为因特网视频应用中最重要的技术之一。

不过，客户端如何监测网络状态和评价 QoE，如何防止播放时缓存器的下溢，在已知当前网络可利用带宽的条件下，如何选择自适应集中的数据段来优化 QoE（获得最好播放质量、最小质量波动和最小起始延时等），以及如何在 TCP 拥塞机制（见 9.2.1 节）之上进行流速率的自适应调整等仍是需要解决的问题。

8.8.3　MPEG DASH 标准

如前所述，MPEG DASH 标准只规定 MPD 的格式和数据段的格式，以使因特网上有效和高质量的视频应用成为可能。

1. 媒体演示描述（MPD）格式

MPD 是一个 XML 文件。图 8-67 以图 8-65 为例给出 MPEG DASH 标准的 MPD 的格式，它具有层次化的结构。“周期”这一层的描述信息包括周期的起始时间、持续时间、所包含的自适应集等。自适应集包含同一节目的数条媒体流，每一条称为一个 Representation，这一层的描述信息给出每一条流的参数，例如连续的段的列表。最下一层是段的描述信息，它给出段的持续时间、序列号、起始时间和 URL 等。对于实况转播，MPD 还给出段的可用性、起始和结束时间、节目开始的大概时间，以及固定或者可变的段持续时间等。

2. 段格式

一个段是响应 DASH 客户端 HTTP GET 或 Partial HTTP GET 请求的实体，段的长度可为 1～10 秒，或者长到 10 秒～2 小时。一个媒体流的第一段可能是初始段，包含初始化 DASH 客户端解码器所需要的信息。每一个段有一个唯一的 URL（可能包括字节范围）、一个索引和显式或隐式的起始时间及持续时间。对于实时转播，还包括段存活的起始和中止时间。每一个段包含至少

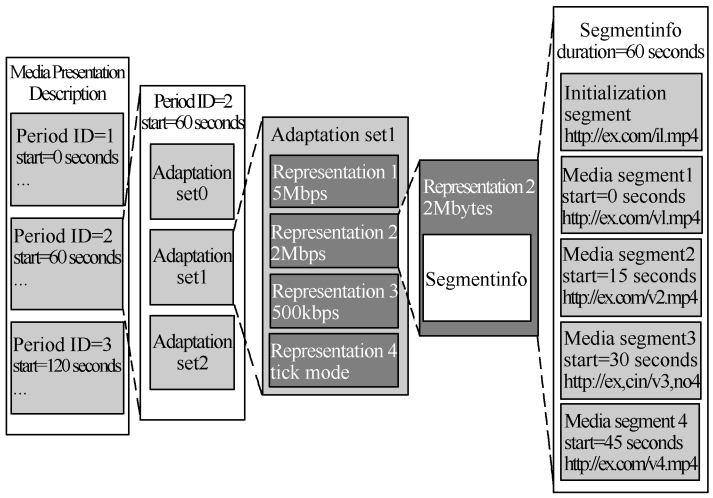


图 8-67　MPD 层次数据模型

一个随机接入点(I 帧或 SP/SI 帧),从该点开始播放或切换流不需要之前的数据就可以进行解码。段的持续时间可以是固定的,也可以是变化的。在实况转播时,后一个段的持续时间可以由前一个段所携带。客户端根据 URL 索取段;根据起始时间和持续时间将段重组成流。

从图 8-67 右侧可以看出,在那个例子中,段是放在 MP4 文件里的。DASH 为 ISO 的基础媒体文件格式(**BMFF**)和 MPEG2 传输流(**TS**)定义了存放段的容器的格式。而 MP4 实际上是 BMFF 的一个实例。由此可见,DASH 客户端播放器能够播放复用或未经复用的码流。

从 8.6.4 节我们知道,MP4 文件的基本单元是"盒"(box)。DASH 定义了一种利用**段索引盒**来下载子段的方法。索引盒给出子段和段中随机接入点的持续时间和字节偏移量的信息。DASH 客户端可以根据这些信息通过 Partial GET 请求下载这些子段。段的索引信息可以放在一个盒里并置于段的开始[见图 8-68(a)],也可以分散在许多盒中。图 8-68 的给出了几种分散放置的方法:分层式、纯链式和混合式。图中深色块代表索引盒,浅色块代表子段,箭头表示出根据箭尾的索引信息可以下载的子段。分散放置索引的好处是,避免段开始处有过多的辅助信息,以减小下载的起始时延。

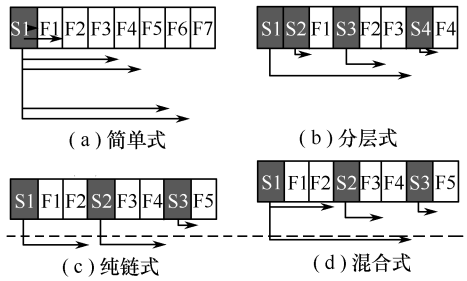


图 8-68　段的索引信息

每一个段中还可以包含通用协调时间 UTC(Coordinate Universal Time),使客户端可以控制自己的时钟漂移。相同的媒体内容可以存储在多个 URL,即多个服务器或 CDN 上,客户端可以从其中任何一个下载,从而最大化自己可利用的网络带宽。

3. 其他

DASH 目前定义了 5 种范畴:MPEG2 TS 的简单和主(Main)范畴,以及 ISO BMFF 的点播、实时转播和主范畴。每一种范畴规定了 MPD 和段格式中的一些限制。

最后,作为客户端的选项,DASH 定义了 QoE 参数测量和向网络提供商的服务器反馈测量结果的格式和协议。主要的参数包括:HTTP 请求/应答事务、切换事件、平均吞吐量、起始播放延时、缓存器充满程度、播放周期的列表和 MPD 信息等。DASH 还定义了一个对段或子段加密的通用标准,它支持客户端采用不同的数字版权管理(DRM)机制。

习 题 八

1. 说明 ISO OSI 的 7 层模型的每一层的功能，分层模型的优点和缺点各是什么？
2. 假设在一个多媒体系统中，采用 UDP 传输实时媒体数据，并且希望传输效率尽可能地高。在你所使用的计算机环境中，分别测试长度为 256、512、1024、2048、4096 和 8192 字节的消息的平均传输速率，并解释原因。
 （提示：观察你所在的网络的最大传输单元 MTU 的大小。）
3. 查阅 RTP/RTCP 协议，了解在默认情况下 H.264 视频流所对应的格式文件和轮廓文件的有关规定。根据 RTP 报文头中的哪一部分信息可以确定该报文是否是某一帧的起始点？你还可以从报文头中了解到哪些有关视频码流的信息？
4. 据统计，目前在因特网（IPv4）上传输的包平均包长为 355B，问：
 （1）当包格式改换成 IPv6 后，平均包长为多少？
 （2）在使用 IPv4、且没有包头压缩时 RTP/UDP/IPv6 包头引入的开销为多少（以百分比表示）？如果每个包的包头压缩至 2 B，引入的开销又是多少？
5. MPEG2 码流中，I、P 和 B 帧 PES 包头中的 PTS 和 DTS 是否相同？
6. 设视频和音频流经 MPEG 压缩后速率分别为 2.5 Mb/s 和 256 kb/s，将它们转换成 TS 流，视频和音频流分别增加 7%和 20%的包头开销；若它们经 RTP/UDP/IP/Ethernet 传输，试计算最终所需要的最小信道带宽。
7. 在一个平均误码率为 10^{-4} 的信道中，假设数据包只要出现错误就被丢弃，如果使包丢失率不大于 10^{-1}，求最大包长。
8. 在 H.323 域中，网关、MCU 和网守的功能是什么？为什么一个域中必须有一个网守，而可以没有 MCU？参照图 8-20 说明 H.323 终端的通信建立过程，及实时媒体数据、应用数据、通信控制信号、信令信号各采用什么协议，为什么？
9. 在图 3-20 的系统中，视、音频分别在 2 个 RTP 会话中传输，（1）说明在发端和收端是如何实现视、音频的流间同步的；（2）参考 RTP/RTCP 报文格式，提出一种在接收端计算平均丢包率的方法。
10. 画出 H.323 终端和 3G-324M 终端相互通信的连接示意图，并具体列出网关应该具备的功能。
11. 从控制功能和传输可靠性两个方面，对 H.242 和 H.245 进行比较。
12. 分别说明在 H.323 和 SIP 的视听系统中，能力集协商是通过什么协议和如何进行的。
13. 说明组通信和多点通信这两个概念的联系和区别。
14. 在一个基于 RTSP 的交互式 VOD 系统中，节目的快进、快退等操作是如何实现的？在 DASH 系统中又是如何实现的？
15. 当若干个相对独立运行的用户（设备、或进程）共享某种资源（传输介质、CPU 时间、存储器以及对某段数据或文件的访问权等）时，通常有如下 4 种基本的处理方法：
 （1）选择一个用户作为主站集中控制资源的分配权；
 （2）建立一个令牌和令牌传递的规则，掌握令牌者占有资源；
 （3）各用户在需要时自由地申请资源，为了防止用户之间的碰撞，将资源加锁。在某个时刻，一把锁只能由一个用户打开；
 （4）允许用户在自由申请资源时发生碰撞，设立检测碰撞和从碰撞状态下恢复正常的机制。
 试从资源的利用率、公平性、可靠性和速度等方面对以上方法加以比较，并对每种方法举一你知道的实例。在用户和资源处于同一台机器内和处于分布式环境中两种情况下，上述讨论结果是否相同？

参 考 文 献

[1] D. D. Clark, et al., "Architectural Considerations for a New Generation of Protocols," Proc. ACM SIGCOMM, 1990, 200-208

[2] H. Khlifi and J-C Gregoire, "Estimation and Removal of Clock shew from Delay measures," proc. IEEE Inter. Conf. on Local Computer Networks, 2004, 144-151

[3] S. T. Kim, et al., "Performance of SCTP for IPTV Applications," Inter. conf. on advanced communication

Technology,2007,2176-2180
[4] C. A. Ellis et al. ,"Groupware:Some issues and experiences,"Comm. ACM,1991,Vol. 34,No. 1,39-58
[5] B. Wang,et al. ,"Multimedia streaming Via TCP:An Analytic Performance Study",Proc、ACM Multimedia,2004.
[6] M. Karcgewicg and R. Kurceren,"The SP-and SI-frames Design for H. 264/AVC,"IEEE Trans. CSVT ,Vol. 13,No. 7,2003,637-644
[7] J. Y. Chung,et al. ,"Scheduling periodic jobs that allows enprecise results,"IEEE Trans. on Computers,Vol. 19,No. 9,1990,1156-1173
[8] P. Shenoy and H. M. Vin,"Efficient striping techniques for maltimedia file servers,"Performance Evaluation,Vol. 38,No. 3-4,1999,175-199
[9] J. E. Wang and P. S. Yu,"Fragmental proxy caching for streaming mutimedia objects,"IEEE Trans. on maltimedia,Vol. 9,No. 1,2007,147-156
[10] WAP "UAProf specification," Oct. 2001, http://www. wapfoum. org/tech/terms. asp? doc = WAP-248-UAProf-20011020-a. pdf
[11] 3GPP TS 26. 244,"Transparent end-to-end packet-switched streamming service(PSS):3GPP file format(3GP)(Release 6),"http://www. 3gpp. org/ftp/specs/archive/26_series/26. 244/26244-630. zip
[12] 3GPP TS 22. 146,"Technical specification group services and system aspects;multimedia broadcast/multimedia service;stage 1,"http://www. 3gpp. org/ftp/specs/archive/22_series/22. 146/22146-660. zip
[13] ISO/IEC 14496-15 Information technology-coding of audio-visual objects——part 15:Advanced video coding (AVC)file format,2004

第 9 章　视频数据的分组传输

9.1　概述

在传统的通信系统中，连续媒体信息（模拟或数字的语音和电视信号）是通过电路交换的网络传输的，数据则通过分组交换的网络传输。而在当前的通信系统中，我们已经从第 7、8 两章中看到，分组交换的 IP 网将在多媒体通信中占据主导的地位。以语音数据分组传输为基础的 IP 电话已得到越来越广泛的应用，视频分组传输的应用也在不断发展中。因此本章将对视频数据在分组传输中遇到的问题加以专门的讨论。

从第 7 章我们已经知道，QoS 保障对于视频在分组网上的传输是十分重要的。由于对 QoS 或 QoE 满意程度的最终评价者是人，因此在分布式多媒体系统中，QoS 的保障不只是多媒体信息传输对网络的要求，而且是一个**端到端**的问题，即从一个终端的应用层到另一个终端的应用层的整个流程中的各个环节均应该具备 QoS 的保障。例如，当播放从远端数据中心传送来的音频和视频数据流时，只有从远端数据库提取、经传输网络传送、直到终端播放的整个过程中媒体流都得到 QoS 的保障时，才能获得满意的播放质量。虽然各种分组网络对 QoS 的保障近年来有不同程度的进展，但还不能满足实时媒体应用的要求，还需要在终端采取一些 QoS 的保障措施。因此本章讲述的重点将是**应用层 QoS 控制**，这包括**应用层拥塞控制**和**应用层差错控制**，其中差错控制又分为在传输层进行的控制和在编码层进行的控制。

9.2　应用层拥塞控制

本节讨论在尽力而为的网络上当发生拥塞时，多媒体终端如何调整自己的发送速率，以避免网络因拥塞而崩溃，同时尽可能地保持高的网络带宽利用率。

9.2.1　TCP 拥塞控制

我们在 8.1.2 节中介绍了 TCP 的流量控制机制，它使发送速率与接收端缓存器的余量相适配；而**拥塞控制**则是对窗口流量控制的补充，它关注的是发送速率与网络当前负荷情况的匹配，即网络负荷轻时 TCP 流可以占用相对多的带宽，而拥塞时则需降低发送速率。TCP 拥塞控制也采用窗口机制，每一个 TCP 连接除了拥有一个流量窗口 Ws 外，还有一个拥塞窗口 W_c，只有在两个窗口都打开时才能传送数据。

图 9-1 给出 TCP 数据传输中拥塞窗口 W_c 的打开过程。TCP 数据传输的开始阶段称为**慢起始**（Slow start），由于不了解当前网络的占用情况，发送端 TCP 首先只发送一个长度等于本次通信约定的最大段长的数据段 **MSS**（Maximum Segment Size），然后启动这个段的重传定时器并等待 ACK。如果超时还未收到 ACK，则重传这个数据段；否则将 W_c 增为 2 倍 MSS。如果这两个数据段的 ACK 都收到，W_c 增为 4MSS，等等。从图看出，在慢起始阶段 W_c 的增长是很快的。当整个慢起始阶段网络没有拥塞、W_c 达到预先设定的慢起始的门限值时，则进入稳态阶段，称为**拥塞避免阶段**。在这个阶段中为了避免拥塞，每收到一个 ACK，W_c 增加 $1/W_c$，也就是说收到 W_c 中所有数据段的 ACK 后，W_c 增加一个 MSS。这个阶段一直持续到 W_c 达到第二个门限（预定的流量上

限值)。自此以后,W_c 保持不变,我们称为 W_c 完全打开,TCP 数据流的流量由流量窗口 W_s($W_s<W_c$)控制。

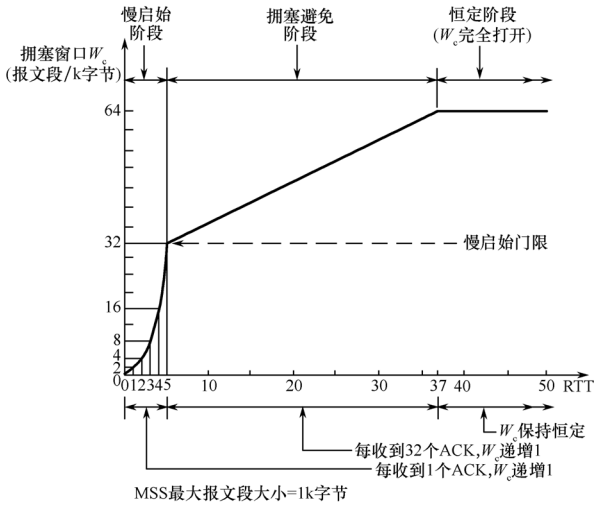


图 9-1　TCP 拥塞控制窗口机制

图 9-1 表示的是网络为轻负载时的情况。如果网络负载加重,就会开始出现丢包的现象。假设由传输差错引起的随机报文丢失可以忽略(≪1%),这个假设在有线网络上是成立的,那未所有检测到的报文丢失都是由于线路(路由器)拥塞引起的。当 TCP 采用称为**快重传**的机制时,即收端 TCP 接收到一个失序的报文段,它必须为已接收的最近一个按序到达的报文发出 ACK。在缓存区中空缺的报文被填补上之前,收端 TCP 必须不断地为每个收到的报文重复这个 ACK,直到空缺填补完成,才为此时收到的所有按序排列的报文发送一个累积的 ACK。因此,当发端 TCP 连续收到具有相同确认号的 ACK 时,可以推测网络有拥塞的迹象,确认号之后的一个报文段很可能丢失了。在 TCP 差错控制中规定,连续 4 次收到具有相同确认号的 ACK(三次收到副本),或者重传定时器超时,则认为报文丢失。前者认为拥塞状况是轻度的;而后者则认为拥塞情况是严重的。在发现轻度拥塞时,TCP 拥塞机制启动,W_c 减低至它的 1/2(而不是降一个 MSS),然后窗口大小线性递增,这称为**快恢复**。图 9-2 给出两次检测到轻度拥塞时 W_c 的变化过程。如果检测到重度拥塞(超时),W_c 重置为一个 MSS,并且以慢起始方式重新获取带宽。

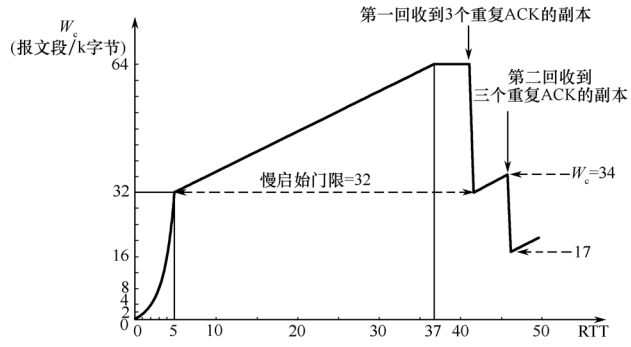


图 9-2　TCP 拥塞控制(轻度拥塞)

从以上分析看出,TCP 在试探性地获取带宽时采取了拥塞窗口增加一个常数(拥塞避免阶段)的策略,而检测到拥塞时则采取窗口大小除 2 的策略。这种调整策略称为**加性增加乘性降低 AIMD**(Additive Increase, Multiplicative Decrease);这种通过改变窗口大小来改变流量的方法则称为**基于窗口的方法**。

9.2.2　TCP 流量模型和 TCP 友好性

1. TCP 流量模型

TCP 流量模型给出在网络拥塞条件下对一个 TCP 流的流量估值。假设网络处于稳态且忽略超时的情况下,经典的 TCP 流量模型如下:

$$T=\frac{\alpha \mathrm{MSS}}{\mathrm{RTT}\sqrt{p}} \tag{9-1}$$

式中,MSS 和 RTT 分别为数据段的平均长度和平均双程延时,p 为包(数据段)丢失率,α 为常数,通常取 $1.5\ \sqrt{2/3}\approx 1.22$。

由于没有考虑超时,因此在网络拥塞较严重、例如丢包率大于 0.02 时,利用上述模型估计的

TCP 流量与实际值有较大的差距。此时，一个更为精确的模型为[1]：

$$T = \min\left\{\frac{W_m \cdot \mathrm{MSS}}{\mathrm{RTT}},\ \frac{\mathrm{MSS}}{\mathrm{RTT}\cdot\sqrt{\frac{2bp}{3}} + \mathrm{RTO}\cdot\min\left[1,3\sqrt{\frac{3bp}{8}}\right]p(1+32p^2)}\right\} \quad (9\text{-}2)$$

式中，W_m 为拥塞窗口的最大值；b 为每一个 ACK 所确认的包数，当 TCP 使用延时 ACK、且接收两个包返回一个 ACK 时，$b=2$；当 TCP 不使用延时 ACK 时，建议 $b=1$；RTO 为规定的重传超时值，通常取 RTO≈4RTT，或者 RTO=max(4RTT，1 秒)。

(9-1)式和(9-2)式说明 TCP 流量主要由丢包率、双程延时和包(分段)大小决定。值得注意，这两个模型都假设 RTT 和 p 独立于所估计的流量，即不考虑当前流量的改变对 RTT 和 p 的影响，因此仅适用于大量数据流进行统计复用的网络环境(如因特网)。当一条瓶颈链路只由少数流共享、而拥塞控制机制又可改变流的发送速率时，(9-1)式和(9-2)式的使用就必须当心，因为该链路上的拥塞情况将受到速率调整的影响；拥塞情况的改变又反过来影响对流量的估值。

2. TCP 友好性

在尽力而为的 IP 网上除了 TCP 流以外，还存在采用其他传输协议的数据流，例如采用 UDP 的视/音频实时媒体流。UDP 没有拥塞控制机制。当网络发生拥塞时，TCP 流各自降低其流量，而 UDP 流仍占有原来的或更宽的带宽，这显然是不公平的。这种不公平性可能导致 TCP“饿死”，严重时甚至产生网络的“拥塞崩溃”。为了解决这个问题，就需要在多媒体终端增加对使用 UDP 的实时媒体流的应用层拥塞控制机制。

好的拥塞控制机制应当保证在共享链路上的所有流，无论它们采用何种传输协议，都公平地分享该链路的带宽，即具有**协议公平性**。由于目前 IP 网上 75%～85%的包流都是 TCP 流，因此非 TCP 的实时媒体流的拥塞控制机制保持与 TCP 的友好性就显得十分重要。对于什么是 **TCP 友好性**(TCP-friendly)目前没有统一的严格界定，本书将采用如下定义：当且仅当一个非 TCP 流在稳态时占有带宽的长期平均值不大于共享链路上 TCP 流在与它相同条件下的带宽，则称该流是 TCP 友好的。有关非 TCP 流在多播情况下 TCP 友好性的定义，有兴趣的读者可参见[1]。

值得注意，非 TCP 流对 TCP 流保持友好，并不一定意味着在瓶颈链路上所有的 TCP 流和 TCP 友好流都享有相同的带宽。即使同是 TCP 流如果其 RTT 不同，或者经过的瓶颈路由器数目不同，它们的流量也是不同的。

9.2.3 基于速率的拥塞控制

图 9-3 给出在单播情况下非 TCP 流进行拥塞控制的基本框图。利用一定的网络反馈机制发送端检测到网络发生某种程度的拥塞时，以与 TCP 友好的方式自适应地降低自己的速率；当拥塞减轻后，则逐步增加速率，重新获取带宽直至达到应有值。相对于利用窗口调整流量的方法而言，这种方法称为**基于速率的控制方法**。

要实现上述结构有三个基本的问题需要解决，即**拥塞检测**、**速率调整**和**调整周期**的确定。对这些问题的不同解决方式则形成了不同的拥塞控制算法。顺便指出，这里的速率调整与 4.3 节的速率调整目的不同，这里是为了防止网络的拥塞，而 4.3 节讨论的是保持编码器平均输出速率为恒定值。

1. 加性增加乘性降低(AIMD)方法

早期的控制方法通过简单地模仿 TCP 的拥塞控制机制来达到与 TCP 的友好性，例如在比较著名的 **RAP**(Rate Adaptation Protocol)算法中，发端发送的数据包头内带有序列号，收端对接收到的包返回 ACK，发端利用返回的 ACK 中的序列号检测包丢失或超时，同时对 RTT 进行估计。

如果网络没有拥塞(无丢包),发送端像 TCP 在拥塞避免阶段那样将发送速率周期性地增加一个固定值;如果检测到网络拥塞(丢包或超时),则立即将发送速率降低到原有值的 1/2。

根据系统控制论,最佳的调整周期与反馈环路的延时有关。所谓反馈延时是指从速率发生变化到检测到网络响应这个变化的时间,对于像 RAP 这样的基于 ACK 的机制来说,反馈延时等于一个 RTT。因此,RAP 速率调整的周期不能小于一个 RTT,否则会引起网络流量的振荡。

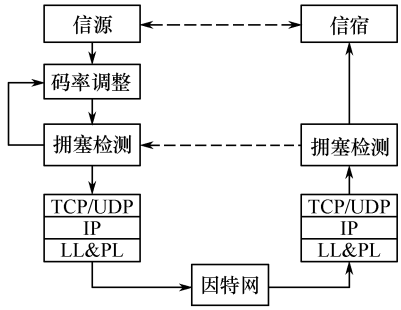


图 9-3　拥塞控制机制的结构

2. 基于流量模型的方法

在基于 AIMD 的方法中检测到一个丢包就将发送速率减半,使得速率的变化呈现出图 9-2所示的 TCP 那样的锯齿形式,这对连续媒体流显然是不合适的。为了使速率变化比较平滑,不少算法根据稳态丢包情况按照 TCP 流量模型(9-1)式或(9-2)式来对速率进行调整,因而被称为**基于模型的方法**。这些方法虽然没有模仿 TCP 的短期行为,但在较长的时间段上仍然是 TCP 友好的。RFC 3448 建议的 **TFRC**(TCP Friendly Rate Control)算法是其中最具代表性的。

TFRC 的速率调整策略如下:

```
If(反馈信息显示接收速率受限
    或者上一次反馈所对应的区间内发送速率已经受限)
    min_rate=max(2 T_recv,W_init/RTT);
  else//典型情况
    min_rate=2T_recv;
If(p>0)//拥塞避免阶段
    按(9-1)式计算 TCP 流量 T_calc;
    T=max[min(T_calc,min_rate),MSS/t_mbi];
  else if(t_now-t≥RTT)//慢起始阶段
    T=max[min(2T,min_rate),MSS/RTT];
    t=t_now
```

在慢起始阶段(见第三个 If),发送端每隔一个 RTT 速率增加约一倍,其中 MSS/RTT 给出慢起始阶段速率的最小值,MSS 为该非 TCP 流的平均包长。当丢包率 $p>0$ 时(见第二个 If),非 TCP 流将速率降至(9-1)式 TCP 流量模型给出的值,以保证与 TCP 的友好性。t_mbi 一般设为 64 秒,它给出发送端长期收不到反馈信息时最大的包间隔,即当 $p>0$ 时,发送端每 64 秒至少发送一个包。我们注意到,在发送速率受限或者链路带宽受限的情况下(见第一个 If),慢起始期间的发送速率并不一定每隔一个 RTT 就能增加一倍,而是限制到 min_rate,min_rate 最高为接收速率 T_recv的 2 倍,或者为初始设定的一个值 W_init/RTT。此外,如果发送端在 4RTT 时间内未收到反馈信息(超时),即网络严重拥塞时,则将发送速率减半,直至停止发送(未表示在上面的伪代码中)。

TFRC 的调整周期也不能小于一个 RTT。TFRC 发送速率的变化比较平滑,有较高的带宽利用率,且与 TCP 是"相对公平"的。所谓相对公平在这里指的是,其发送速率与同样条件下的 TCP 流相比不超过 2 倍。为速率变化平缓所付出的代价则是,TFRC 对网络带宽变化的响应速度要慢于 TCP。

9.2.4　拥塞检测

从前面的讨论可以看到,网络拥塞通常是通过丢包率来衡量的,因此对由于拥塞而引起的丢包

率的准确估计是拥塞控制中的一个重要问题。

1. TFRC 的拥塞检测

TFRC 要求接收端在一个 RTT 内至少向发送端反馈一次信息。TFRC 发送包头上带有序列号 i 和该包发送时刻的时间戳 t_i，以及发送端估计的 RTT 值等；反馈信息包头则带有收到的最后一个包的时间戳 t_i、从收到该包到发出反馈信息包的延时 t_delay、上一个 RTT 中的接收速率 T_recv 和估计的丢包率 p 等，其中 T_recv 在前面讲述的速率调整中曾经用到。

发送端在 t_now 时刻收到反馈包，则当前的双程延时 RTT_sample 为

$$\text{RTT_sample}=(\text{t_now}-\text{t_i})-\text{t_delay} \tag{9-3}$$

双程延时的估计值则为

$$\text{RTT}=\alpha\text{RTT}+(1-\alpha)\text{RTT_sample} \tag{9-4}$$

式中，α 通常取 0.9。

为了滤除网络负载瞬间波动的影响，TFRC 不是通过单个丢包，而是通过平均丢包事件间隔来对网络拥塞情况进行估计的。在一个 RTT 之内发生的一个或多个丢包称为一个**丢包事件**。图 9-4 给出若干丢包事件的间隔，图中小方块表示收到的包，X 表示丢包事件的边界。丢包事件以丢包时刻为边界，如果新的丢包时刻与当前丢包事件的起始时刻之差小于 RTT，则新丢包属于当前丢包事件；反之，则从新丢包时刻开始计为下一个丢包事件。由于一个 RTT（一个丢包事件）中丢失的包的个数不等，所以图中丢包事件间隔长度不等，换句话说每个丢包事件间隔中收到包的个数不等。为了反映当前网络的拥塞状况，我们用加权平均丢包事件间隔 S 来估计丢包率 p。

$$\bar{S}=\sum_{i=1}^{n}S_i\cdot W_i/\sum_{i=1}^{n}W_i \tag{9-5}$$

其中

$$W_i=\begin{cases}1 & (1<i\leqslant\dfrac{n}{2})\\ 1-\dfrac{i-\dfrac{n}{2}}{\dfrac{n}{2}+1} & (\dfrac{n}{2}<i\leqslant n)\end{cases} \tag{9-6}$$

则

$$p=1/\bar{S} \tag{9-7}$$

(9-5)式中 W_i 是对不同丢包事件间隔的加权值（见图 9-4 右边）；S_i 是第 i 个丢包事件的长度（间隔），也就是这个间隔中收到的包数。n 的取值影响 *TFRC* 对网络负载变化响应的速度，通常$n\leqslant 8$。当最后一个丢包事件的间隔（还未结束）大到足以影响平均事件间隔时，n 从最后一个丢包事件间隔开始计；否则，从前一个事件间隔开始计算（图 9-4 所示的情况）。

2. 基于 RTP/RTCP 的拥塞检测

有的拥塞控制算法利用 RTP/RTCP 获得网络状态信息。我们在 8.1.3 节中已经讲述了通过 RTCP 的 SR 和 RR 报告计算丢包率 p 和双程延时 RTT 的方法。但是 RTCP 的反馈周期较长（最少 5 秒），因此这类算法对网络负载变化的响应比较慢。同时标准的 RTCP 能够报告的最小包丢失率有一定限制，在包丢失率很小时，RTCP 报告 $p=0$，此时非 TCP 流会占有过多的带宽。

3. 路由器支持的拥塞检测

前面介绍的拥塞检测方法都是通过在接收端对丢包进行检测来实现的。在某些网络中，路由器可以在输出队列溢出（丢包）之前给出拥塞指示，这称为**显式拥塞指示 ECN**（Explicit Congestion Notification，RFC 3168）。这样的路由器使用主动队列管理方法，**随机早期检测**（Random Early Detection，RED）就是主动队列管理方法的一种。在 RED 中，当一个包送到路由器输出队列时，如果队列的平均长度 L 小于最小阈值 Thmin，则该包进入队列；如果 L 大于最大阈值 Thmax，该包

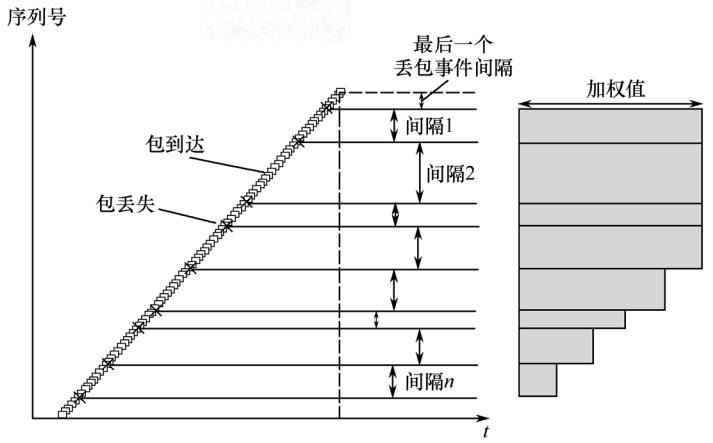


图 9-4　丢失事件加权

被丢弃；如果 Thmin＜L＜Thmax，则从队列中随机选取一个包丢弃。如果被随机挑选的包不将其丢弃、而是将其 IP 头中的一个标志位置位；接收端收到这样的包(ECN_包)后就知道网络已经出现了拥塞的迹象，从而将此信息反馈给发送端，使发送端启动拥塞控制机制。在尚未丢包的情况下就获得拥塞指示，对音、视频这样的对时延和丢包都敏感的应用是很有利的。

图 7-20 曾给出 IPv4 报头中的一个字节的“服务类型”域，在 IPv6 中对应于“业务等级”域，该域的最后 2 比特未用。这 2 比特在 ECN 中被用作 **ECN 域**。该域为 00 时表示不使用 ECN；01 和 10 由发送终端置位，表示终端的传输层协议可以感知 ECN；11 由路由器置位，用作拥塞指示。在终端的拥塞控制机制中，对于带有 ECN 指示的包按丢包一样处理。值得注意，实现 ECN 必须路由器和终端均支持相应的协议。

4. 端到端的可用带宽测量

通过对当前可用带宽的测量可以从另一个角度反映网络负载的情况，从而为端到端的拥塞控制提供指示。测量的方法分为主动和被动两种。**主动测量**(Active Probing)通过在发送端发送专门的探测包，然后在接收端分析包到达间隔和/或单程延时等参数的变化规律来得到对当前网络可用带宽的估值；**被动测量**则利用正在传输的媒体流来完成相应的测试。值得注意，这里的可用带宽测量是指在终端之间(端到端)进行的；在网络节点(路由器、交换机等)处进行的主动或被动测量虽然可以得到可利用链路带宽的较为准确的估计，但是这些估值很难被终端所获得，因此难以在拥塞控制中使用。有兴趣对可用带宽测量做深入了解的读者可以参见文献[2,3]。

9.2.5　无线网络中的拥塞控制

在有线网络中，TCP 和非 TCP 流的拥塞控制机制，如前所述，都假设所有的丢包是由于路由器的缓存器溢出，即拥塞而引起的。但在无线网络中，情况却复杂得多。由于噪声干扰、多径衰落等原因，无线链路的随机错误和突发错误率很高，一般在 $10^{-3}\sim10^{-2}$。此外，当移动终端从一个小区漫游到另一个小区时，可能因为切换而产生信号的短时中断。而在无线自组织网络中，终端自由移动形成的网络拓扑动态变化可能引起路由的失败，等等。上述这些原因都会产生单个或连续多个数据包的传输丢失，而此时并不存在路由器缓存器的拥塞问题。如果仍然按照有线网络上的拥塞控制策略去降低源端的发送速率，将极大地降低无线链路的带宽利用率。

针对改善 TCP 在无线网络上的性能，研究者已经提出了许多方法。这些方法大致可以分为如下几类：

(1) 链路层措施。链路层将报文划分为更小的段[根据(8-4)式，小的包长有小的丢包率]并引

入自动重传机制，以降低无线链路的传输错误，使得 TCP 拥塞控制的启动主要由拥塞丢包引起。但这种方法在 IP 包被加密的情况下，链路层不能识别 TCP 流而无法使用。

(2) 显式指示。接收端显式地通知发送端包丢失的原因(拥塞还是误码)，类似于 ECN，这往往是在中间路由器或基站的帮助下完成的。发送端根据丢包原因，决定是否启动和如何启动拥塞控制机制。

(3) 端到端的措施。这类方法维持 TCP 端到端连接的特性，而将网络视为黑盒子。将所有丢包视为拥塞、且将拥塞窗口乘性降低(MD)是 TCP 在无线网络上性能降低的原因，因此根据区分丢包成因(拥塞、误码或断连)的方法和拥塞控制策略及参数(拥塞窗口、超时定时器、慢启始门限等)调整方法的不同，形成了许多种无线网上 TCP 的改进协议。一些协议通过分析信号功率的大小来判断线路的短时中断；通过分析双程延时、延时抖动或包到达时间间隔等变化来区分拥塞与误码；然后根据检测结果决定是否启动 MD。另一些协议试图控制 MD 的程度，尽管对每一个丢包都启动 MD，但调整拥塞参数使吞吐量仍然维持在较高的水平上。有许多协议则直接让发送速率与发送端估计的当前网络可利用带宽相匹配。

(4) 分段处理。对于有线和无线混合的网络，可以将二者分开处理；利用边界基站接收、缓存有线网络的 TCP 包，并返回 ACK。这种方法不能维持 TCP 的端到端连接的特性。

对非 TCP 流在无线网络上的拥塞控制，目前研究还很少，但各种无线 TCP 中采用的方法有一定的借鉴作用。例如，分析单程延时和包到达间隔等的变化规律来区分丢包是由拥塞还是由误码而引起的，通常认为当拥塞出现时，单程延时呈单调上升的趋势；而误码则影响包的到达时间间隔，或者误码引起的丢包比拥塞的引起的丢包更具有随机性。然后，只在拥塞引起丢包时启动拥塞控制机制。

9.3 传输层差错控制

本节介绍通用的传输层差错控制技术。

9.3.1 自动重传技术(ARQ)

自动重传请求(Automatic Repeat reQuest，ARQ)是一种基于反馈的差错控制机制。在这种机制中，数据流分割成段，每段加上少量的校验比特后发送出去；当接收端通过校验比特检测到数据段中有错误时，则请求重新发送该数据段。ARQ 是物理层和数据链路层经常采用的差错控制方法；在传输层上采用 ARQ 时，可以利用低层协议检测包丢失，而不一定在包中加入检验比特。

下面介绍几种基本的 ARQ 协议。

1. 停止和等待(Stop and Wait)

在这种方式中，发送端发送一个包然后等待 ACK，接收端收到无差错的包后返回 ACK，发送端收到 ACK 后再发送下一个包；如果等待超时仍未收到 ACK，则重发该包。因此，在任何给定时刻只有一个包没有被确认。由于在等待 ACK 期间信道没有被利用，此种方法的数据传输率较低，也较少在实际中应用。

2. 选择性重传(Selective repeat)

发送端连续发送数据包，接收端对每一个接收到的数据包返回一个带有该包序列号的 ACK。如果一个包超时仍未收到 ACK，则重传该包；然后继续原来的发送，即假设已发送的包的最大序列号为 a，则继续从 a+1 开始发送。在选择性重传中，数据包是连续发送的，信道利用率较高，但是接收端所收到的包可能次序颠倒，需要缓存和重新排序。

3. **退回 N(Go-back-N)**

退回 N 方法与选择性重传一样，发送端连续发送数据包，但是在接收端只按顺序接收包；不按顺序到达的包被丢弃，也不返回 ACK。当发现超时时，发送端从超时的数据包开始重新依次发送。因此这种方法既保证了较高的信道利用率，也不需要缓存和重新排序处理。

利用 ARQ 纠正错误至少引入一个 RTT 的延时，因此在实时媒体的传输中需要谨慎使用。只有在重传的数据包能够在它的播放时限之前到达接收端时，使用 ARQ 才有意义。除此之外，在带宽－延时积接近于发送窗口时，ARQ 可能使吞吐量下降。ARQ 适于在点对点的通信中使用，在多播情况下，各接收端的丢包情况不同，此时使用 ARQ 效率会很低。

9.3.2 前向纠错码

1. **基本定义**

纠错编码是基于特定数学的一个技术分支，在本节中将从工程技术的需要出发，力图用最少的数学知识使读者对前向纠错码有一个基本的认识，对其中的许多结论将省略其严格的数学证明。有兴趣进一步了解前向纠错码的读者可参考有关纠错编码的书籍。

发送端在要传送的数据中加入一定的冗余数据，生成特定的码；接收端收到这些码后，通过译码不但能够自动检测错误，而且能够纠正错误，这种码称为**前向纠错码**(Foward Error Correction，FEC)。前向纠错不需要反馈信息，因而适于在视频数据的传输中应用。常用的前向纠错码为**分组码**和**卷积码**两种，本节主要介绍分组码。

分组码编码器将一组输入符号(代表需要传送的数据)加上冗余信息后转换成一组输出符号。这里讲的符号可以是 2 电平(二进制)的，也可以是多电平的。输出符号组中的符号数比输入的多，这两组符号数目之比称为**编码效率**(Code Rate)。例如一个(n,k)分组码，意味着 k 个输入符号，加上$(n-k)$个冗余校验符号，产生 n 个输出符号。

某一种码的码字是输出符号所有可能的组合中的一个限定的子集。如果接收到的符号组不是有效的码字(不在限定的子集之内)，则说明传输中产生了错误。找到与错误的符号组“距离最近”的有效码字就可能纠正传输差错。这里所说的“距离”为**汉明(Hamming)距离**，它用两个码字之间不同的符号的个数来度量。例如有 4 个符号互不相同，则这两个码字之间的汉明距离为 4。如果说某种码的汉明距离为 4，那么意味着该码中任何一对有效码字之间至少有 4 个符号是相互不同的。一般来说，一种码能校正的最多错误个数 $t=(d-1)/2$，其中 d 为**最小码间距离**。

如果一种码通过码的位置的重新排列，或者通过符号的互易(例如对二进码而言，将 0 换成 1，将 1 换成 0)能够变换成另一种码，我们称这两种码是**等效的**。此外，如果一个 FEC 编码器符合线性系统的定义，那么该编码器称为**线性编码器**。

2. **有限域**

为了理解码的运算，有必要介绍一些有限域(或称为伽罗华域)的知识。一个**有限域**由一个有限元素的集合和在这个集合上定义的加法和乘法运算构成，域内进行加法或乘法运算的结果仍是域内的一个元素。域内有一个 0 元素和一个 1 元素，对于域内的任何一个元素 a 满足

$$\begin{aligned} a+0&=a \\ a\cdot 1&=a \end{aligned} \tag{9-8}$$

a 的加法的逆为$(-a)$，并且

$$a+(-a)=0 \tag{9-9}$$

a 的乘法的逆为a^{-1}，并且

$$a\cdot a^{-1}=1 \tag{9-10}$$

法国数学家伽罗华(Galois)已经证明,有限域的大小 p(称为"**阶**"),记为 $GF(p)$,必须为质数,或质数的幂。当 p 为质数时,有限域由模 p 加法和模 p 乘法运算的余数的集合构成。例如 $p=7$ 时,由模 7 的加法和乘法构成的域为{0,1,2,3,4,5,6}。在域中如果任何一个非零元素都可以用某一个元素的幂来表示,那么该元素称为**本原元**(Primitive Element)。每个域中至少有一个本原元。例如在上述例子中,3 是该域的本原元,其他元素可以表示为 $3^0(=1)$, $3^1(=3)$, $3^2(=2)$, $3^3(=6)$, $3^4(=4)$, $3^5(=5)$。

与模 p 加和模 p 乘法相类似,任何两个系数是实数的多项式 $f(x)$ 和 $g(x)$,可以表示成

$$\begin{aligned} f(x) &= q(x)\cdot g(x)+r(x) \qquad (0\leqslant \text{order}\{r(x)\}<\text{order}\{g(x)\}) \\ &\equiv r(x) \qquad \text{Mod}[g(x)] \end{aligned} \tag{9-11}$$

式中,order$\{r(x)\}$ 表示 $r(x)$ 的**阶数**,而 $f(x)\equiv r(x)$,Mod$[g(x)]$ 表示 $f(x)$ 用多项式 $g(x)$ 除后余式为 $r(x)$。

若 m 次多项式

$$f(x)=f_m x^m+f_{m-1}x^{m-1}+\cdots+f_1 x+f_0 \tag{9-12}$$

的系数 $f_i(i=0,1,\cdots,m)$仅取 $GF(q)$ 中的元素,则称 $f(x)$ 是***GF(q)* 上的多项式**。例如 $f(x)=x^3+x+1$,其系数只取 0 和 1,因此 $f(x)$ 是 $GF(2)$ 上的多项式。如果 $GF(q)$ 上的某个多项式除了常数和本身外,不能再被 $GF(q)$ 上的其他多项式除尽,则该多项式称为**既约多项式**。若 $g(x)$ 是 $GF(q)$ 上的 c 次既约多项式,那么所有次数小于 c、系数取自 $GF(q)$ 的多项式的集合,在模 $g(x)$ 相加和相乘运算下,构成***GF*(q^c) 的有限域**。

当 $GF(p)=q^c$ (q 为质数)时,有限域的非零元素由下列方程的根构成

$$x^{q^c-1}+1=0 \tag{9-13}$$

以 $GF(8)=2^3$ 的情况为例,(9-13)式分解成

$$x^7+1=(x^3+x+1)(x^3+x^2+1)(x+1)=0 \tag{9-14}$$

上式中 $GF(2)$ 上的 c 阶既约多项式(x^3+x+1) 和(x^3+x^2+1) 称为**本原多项式**。选择任何一个本原多项式都可以构成有限域。例如,在模(x^3+x+1)运算下,$GF(8)$ 的有限域为{0,1,x,x^2,$x+1$,x^2+x,x^2+1,x^2+x+1}。

域的元素可以用域的本原元 α 的幂来表示。在上面 $GF(8)$ 的例子中,$\alpha=x$,在模(x^3+x+1)运算下,非零元素可以表示为:$\alpha^0=1$,$\alpha^1=x$,$\alpha^2=x^2$,$\alpha^3=x+1$,$\alpha^4=x^2+x$,$\alpha^5=x^2+x+1$,$\alpha^6=x^2+1$。

域的元素也可以用数字形式表示,通常有两种表示方法。一种是幂表示法,即直接给生成元(本原元)的幂指定对应的二进制码,在这里,零元素通常指定为 2^c-1。另一种是组合表示法,即将域内的元素表示成由 α 的幂所构成的多项式,并指定 000,001,010 和 100 分别代表 0,α^0,α^1 和 α^2,$\alpha^n(n>2)$ 的元素则用次数小于 2 的 α 的幂的和表示,并由此得到相对应的二进制表达式。例如,$\alpha^3=\alpha+1(011)$,$\alpha^4=\alpha^2+\alpha(110)$,$\alpha^5=\alpha^2+\alpha+1(111)$,$\alpha^6=\alpha^2+1(101)$。

3. 线性分组码

线性分组码通过下式生成:

$$\boldsymbol{v}=\boldsymbol{u}\cdot G \tag{9-15}$$

式中,$\boldsymbol{u}$ 为输入的信息符号组,$\boldsymbol{v}$ 为输出分组码,$\boldsymbol{G}$ 为一个 $k\times n$ 的**生成矩阵**,其中 k 为输入符号的个数,n 为输出码字的长度。长度为 k 的信息组共有 2^k 种,通过编码器后,对应的码字也应该只有 2^k 个,但是长度为 n 的码字可能有 2^n 个,因此分组码的编码问题就是定出一套规则,从 2^n 个 n 位码中选出 2^k 个作为**有效码字**,其余的 2^n-2^k 个称为禁用码字。k 个线性独立的有效码字的线性组合仍产生另一个有效码字。

对于任意一种分组码，其生成矩阵 $\boldsymbol{G}$ 都可以经过有限次的初等变换化为

$$\boldsymbol{G}=[\boldsymbol{I}_k|\ \boldsymbol{P}] \tag{9-16}$$

其中 I_k 为 $k\times k$ 的单位矩阵，$\boldsymbol{P}$ 为一个 $k\times(n-k)$ 阶的矩阵。用 $[\boldsymbol{I}_k|\boldsymbol{P}]$ 生成的分组码与用 $\boldsymbol{G}$ 生成的码是等效的，而且用 $[\boldsymbol{I}_k|\boldsymbol{P}]$ 生成的码的前 k 位与输入信息组完全相同，只在后面附加了 $(n-k)$ 个校验位。这种信息组未经变化而出现在码字中的码称为**系统码**。

设 $C=[C_{n-1}\cdots C_1C_0]$ 为任意一个有效码字，若 $(n-k)\times n$ 阶矩阵 $\boldsymbol{H}$ 满足

$$C\cdot\boldsymbol{H}^{\mathrm{T}}=0 \tag{9-17}$$

则称 $\boldsymbol{H}$ 为该 (n,k) 码的**校验矩阵**(Parity Check Matrix)。由(9-15)式可知，(n,k) 码的所有 2^k 个码字都可由 $\boldsymbol{G}$ 的行的线性组合生成，因此 $\boldsymbol{G}$ 中的每一行也是 (n,k) 码的有效码字。

根据(9-17)式我们有

$$\boldsymbol{G}\cdot\boldsymbol{H}^{\mathrm{T}}=0 \tag{9-18}$$

如果 G 是系统码的生成矩阵，则由(9-16)和(9-18)式得到 $\boldsymbol{H}=[-\boldsymbol{P}^{\mathrm{T}}\mid I_{(n-k)}]$。因此，对于系统码而言，由 $\boldsymbol{G}$ 可以很容易推出 $\boldsymbol{H}$，反之亦然。

线性分组码的解码过程如下。

(1) 检查接收到的符号组 $\boldsymbol{v}$ 是否使下列方程成立：

$$\boldsymbol{v}\cdot\boldsymbol{H}^{\mathrm{T}}=0 \tag{9-19}$$

如果(9-19)式成立，则 $\boldsymbol{v}$ 是有效码字。

(2) 如果上式不成立，则发生误码。假设接收的符号组 $\boldsymbol{v}=\boldsymbol{c}+\boldsymbol{e}$，其中 $\boldsymbol{c}$ 为有效码字，$\boldsymbol{e}$ 为错误图形(Error Pattern)，那么

$$\boldsymbol{v}\cdot\boldsymbol{H}^{\mathrm{T}}=(\boldsymbol{c}+\boldsymbol{e})\cdot\boldsymbol{H}^{\mathrm{T}}=\boldsymbol{e}\cdot\boldsymbol{H}^{\mathrm{T}}\equiv S \tag{9-20}$$

式中 S 称为 $\boldsymbol{v}$ 的**伴随式**(Syndrome)，它完全由错误图形确定。根据 S 和校验矩阵 $\boldsymbol{H}$ 可以检测错误的位置从而纠正错误。

4. 汉明码

我们在本节以比较简单的汉明码(Hamming Codes)作为例子来理解线性分组码的编、解码过程。任何一个 (n,k) 线性分组码有 $(n-k)$ 个校验位，在二进制的情况下，这 $(n-k)$ 个校验位可以构成 $(2^{n-k}-1)$ 个互不相同的非全零的 $(n-k)$ 比特图形。将这 $(2^{n-k}-1)$ 个不同的比特图形作为列所组成的 $(n-k)\times(2^{n-k}-1)$ 矩阵则为汉明码的校验矩阵 $\boldsymbol{H}$。例如对于一个(7,4)二进汉明码，$(n-k)=3$，有 $(2^3-1)=7$ 种非全零的 3 比特图形：(001)，(010)，(011)，(100)，(101)，(110)和(111)。用这 7 个非零的 3 比特图形作列，构成系统码的校验矩阵 $\boldsymbol{H}$ 为

$$\boldsymbol{H}=\begin{bmatrix}1&1&1&0&1&0&0\\0&1&1&1&0&1&0\\1&1&0&1&0&0&1\end{bmatrix}=[\boldsymbol{P}^{\mathrm{T}}\mid\boldsymbol{I}_3] \tag{9-21}$$

由 $\boldsymbol{H}$ 推出系统码的生成矩阵 $\boldsymbol{G}$(模 2 运算)：

$$\boldsymbol{G}=\begin{bmatrix}1&0&0&0&1&0&1\\0&1&0&0&1&1&1\\0&0&1&0&1&1&0\\0&0&0&1&0&1&1\end{bmatrix}=[\boldsymbol{I}_4\mid\boldsymbol{P}] \tag{9-22}$$

汉明码的一个特点是校验矩阵 $\boldsymbol{H}$ 中的第 n 列代表码字中第 n 位的伴随式，可以在译码时根据这个特点检测和纠正误码。例如在上例中，假设输入的信息符号组 $\boldsymbol{u}=[0101]$，经(7,4)二进汉明编码后，输出的分组码 $\boldsymbol{v}=\boldsymbol{u}\cdot\boldsymbol{G}=[0101100]$。假如在传输过程中码字的第 2 位上产生了错误，接收到的码 $\boldsymbol{v}'=[0001100]$。利用(9-20)和(9-21)式计算伴随式 S：

$$S=\boldsymbol{v}'\cdot\boldsymbol{H}^{\mathrm{T}}=[111] \tag{9-23}$$

伴随式与(9-21)式所示的校验矩阵中第 2 列相同，因而接收端知道在第 2 位存在误码，将 $\boldsymbol{v}'$ 中第 2 位 0 改为 1，则得到正确的码字 0101。

5. BCH 码

BCH 码(Bose-Chaudhuri-Hocquenghem Codes)是在实际中应用得比较广泛的分组码。

由于域内的非零元素都是(9-13)式的根，因而 c 阶本原多项式在 $GF(q^c)$ 域中可以分解成一次因式的乘积。例如在 $GF(2^3)$ [见(9-14)式]域中则可分解为

$$x^3+x^2+1=(x+\alpha)(x+\alpha^2)(x+\alpha^4) \tag{9-24}$$

$$x^3+x+1=(x+\alpha^3)(x+\alpha^5)(x+\alpha^6) \tag{9-25}$$

$$x+1=(x+\alpha^0) \tag{9-26}$$

当 c 阶本原多项式的一个根 β 找到以后，这个多项式的其他根由下式给出：

$$\text{根 } \beta \Rightarrow \text{根 } \beta^q,\ \beta^{q^2},\ \beta^{q^3},\ \cdots,\ \beta^{q^c} \tag{9-27}$$

式中，指数是按模(q^c-1)计算的。例如，α^3 是 x^3+x+1 的一个根，那么 α^6 和$(\alpha^3)^4=\alpha^5$ 也是它的根。

我们可以定义一种码，该码由在 $GF(q^c)$ 中具有某些特定根的多项式构成。例如，如果将根为 α 的多项式作为码的码字(称为**码多项式**)，那么所有的码字都能被本原多项式 x^3+x^2+1 整除(参照(9-24)式)。换句话说，本原多项式可以作为码的**生成多项式**。如果用根 β 来定义码，那么所有的码字(码多项式)v 均满足

$$v(\beta)=v_{n-1}\beta^{n-1}+\cdots+v_2\beta^2+v_1\beta^1+v_0\beta^0=0 \tag{9-28}$$

同理，如果生成多项式 $g(x)$ 以 $\beta_1,\beta_2,\cdots,\beta_j$ 为根，则所有的码多项式也必须以这些元素为根，写成矩阵形式为

$$\begin{bmatrix} v_{n-1} & \cdots & v_2 & v_1 & v_0 \end{bmatrix} \cdot \begin{bmatrix} \beta_1^{n-1} & \beta_2^{n-1} & \cdots & \beta_j^{n-1} \\ \vdots & \vdots & & \vdots \\ \beta_1^1 & \beta_2^1 & \cdots & \beta_j^1 \\ \beta_1^0 & \beta_2^0 & \cdots & \beta_j^0 \end{bmatrix} = 0 \tag{9-29}$$

具有上面根矩阵(右边的一个)形式的矩阵称为 **Vandermonde 矩阵**。将上式与(9-19)式相比可知，上式中的根矩阵就是该码的校验矩阵的转置 $\boldsymbol{H}^{\mathrm{T}}$，且 $j=n-k$。如前所述，由 $\boldsymbol{H}$ 可以很容易得到 BCH 码的生成矩阵 $\boldsymbol{G}$。

BCH 码由 $GF(q^c)$ 域中 $2t$ 个幂指数连续(即指数逐次递增 1)的根来定义，其码长 $n=(q^c-1)$，最小码间矩 $d=2t+1=n-k+1$。这种码可以校正 t 个错误。实际上汉明码也是一种 BCH 码，它由 2 个连续根来定义，可以纠正一个错误。假设我们选择(9-24)式作为生成多项式，在 $GF(2^3)$ 中它的根为 α,α^2 和 α^4，其中两个连续根为 α,α^2。用这两个连续根生成 BCH 码，根据(9-29)式，它的校验矩阵为

$$\boldsymbol{H}=\begin{bmatrix} \alpha^6 & \alpha^5 & \alpha^4 & \alpha^3 & \alpha^2 & \alpha^1 & 1 \\ \alpha^{12} & \alpha^{10} & \alpha^8 & \alpha^6 & \alpha^4 & \alpha^2 & 1 \end{bmatrix}=\begin{bmatrix} \alpha^6 & \alpha^5 & \alpha^4 & \alpha^3 & \alpha^2 & \alpha^1 & 1 \\ \alpha^5 & \alpha^3 & \alpha & \alpha^6 & \alpha^4 & \alpha^2 & 1 \end{bmatrix} \tag{9-30}$$

将在对有限域的讲解中定义的 α 的二进制表达式代入上式，得到

$$\boldsymbol{H}=\begin{bmatrix} 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 0 & 1 & 1 & 1 & 0 & 1 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \\ 1 & 0 & 0 & 1 & 1 & 1 & 0 \\ 1 & 1 & 1 & 0 & 1 & 0 & 0 \\ 1 & 1 & 0 & 1 & 0 & 0 & 1 \end{bmatrix} \tag{9-31}$$

上面矩阵的后 3 行与前 3 行线性相关，因此

$$\boldsymbol{H}=\begin{bmatrix}1&1&1&0&1&0&0\\0&1&1&1&0&1&0\\1&1&0&1&0&0&1\end{bmatrix} \tag{9-32}$$

上式与(9-21)式完全相同，说明这样生成的 BCH 码即为一个二进制的(7,4)汉明码。

6. RS 码

一般地讲，一个 q 进制的 BCH 码，每个码字的取值在 GF(q) 域上，而它的生成多项式 $g(x)$ 的根却在 GF(q^c) 域(称为**扩展域**)中。如果码字的取值域和码的 $g(x)$ 的根所在的域相同，则称这类 BCH 码为 **RS 码**(Reed-Solomon Codes)。RS 码在数据通信、数字视频传输和存储系统中得到了广泛的应用。

RS 码是一种在 $GF(q)$ 域内的、码长为($q-1$)的多进制码。如果 $q=256$，则意味着 RS 码是 256 进制的，码长为 255。与一般 BCH 码的生成方法一样，RS 码的校验矩阵 $\boldsymbol{H}$ 和生成矩阵 $\boldsymbol{G}$ 可以根据式(9-29)和式(9-18)获得。

在实际应用中，有时很难找到一个有合适码长或合适信息位长度的分组码，此时可以把一个已知的分组码缩短。所谓**缩短码**是指在(n,k)码的 2^k 个码字集合中，挑选前 i 个信息位均为 0 的码字，共有 2^{k-i} 个，组成新的码字的集合。它是原码字集合中的一个子集，该子集组成一个($n-i$,$k-i$)分组码，称为(n,k)的缩短码。例如(204,188)RS 码是(255,239)的缩短码。由于缩短码中的前 i 位的值均为 0，因此发送时不必传送这 i 位，只传送后面的($n-i$)位即可。缩短码的纠错能力至少与原来的(n,k)码相同。对于 RS 码，缩短之后的最小码间距不改变。由于缩短码是去掉原码的 i 个信息位得到的，编码效率较低，纠错能力又不一定比原码强，所以总的说来缩短码的性能要比原码稍差。

9.3.3 前向删除复原码

传统的前向纠错码(FEC)广泛地应用在点对点的通信线路中，从 ISO 7 层参考模型的角度而言，FEC 多应用于网络的低层，如有线网络的链路层、或无线网络的链路层及物理层，用以检测和纠正线路中发生的随机比特错误和突发错误。在分组网(如 IP 网)中，通常低层检测到不可纠正的错误时将包丢弃；同时丢包也可能如 9.2 节所述由于拥塞而产生，在这两种情况下网络高层(如传输层)或者收到正确无误的包，或者包丢失，而且包丢失的位置通过低层协议或其他协议(如 RTP)可以精确地获知。这种信道我们称为**删除信道**(Erasure Channel)，恢复信道删除错误的码称为**删除复原码**(Erasure resilient codes)。删除复原码也基于 9.3.2 节讲述的 FEC 的原理，但与传统的前向纠错码针对较短的比特串进行操作不同，删除复原码是针对数据包进行操作的，因此它也常被称为**包级别 FEC**。

1. 包级别线性分组码

一个数据包可能包括几百甚至几千比特，在这里我们假设将大数据包分成多个数据段，对于每个数据段可以进行有限域上的算术运算。根据(9 -15)式，我们有

$$\boldsymbol{y}=\boldsymbol{G}\boldsymbol{x} \tag{9-33}$$

其中 $\boldsymbol{x}$ 为输入的信息符号组，$\boldsymbol{y}$ 为输出分组码，$\boldsymbol{G}$ 为一个 $n\times k$ 的生成矩阵。生成矩阵可以通过(9-29)式得到。值得注意，在上式中 $\boldsymbol{x}$ 和 $\boldsymbol{y}$ 分别为 $k\times1$ 和 $n\times1$ 的列向量，而在(9-15)式中输入向量 $\boldsymbol{u}$ 和输出向量 $\boldsymbol{v}$ 则分别是 $1\times k$ 和 $1\times n$ 的行向量。显然当上式的 $\boldsymbol{G}$ 为(9-15)式中生成矩阵的转置时，上式和(9-15)式是等效的；也就是说，删除复原码的构成和传统的分组码是相同的。只不过对传统纠错码的讨论往往用多项式表示，而对删除复原码的讨论则多采用矩阵的形式。

对于传统的分组码，由于接收到的码字不知道是否存在错误，也不知道错误发生在哪个比特位上，因此译码时需要通过校验矩阵 $\boldsymbol{H}$ 来检测和纠正误码（见(9-19)式和(9-20)式）。但是在删除信道上接收到的（未被删除的）码字肯定是正确的，因此根据(9-33)式，若生成矩阵 $\boldsymbol{G}$ 的任意 k 行组成的 $k\times k$ 子矩阵 $\boldsymbol{G}_k$ 都是可逆的（分组码的生成矩阵满足此条件），则只要接收到任意 k 组码字就能通过解 k 个方程恢复出原来的信息符号组。这意味着该码可以纠正 $n-k$（即 $d-1$）个删除错误；而相同的码用于前向纠错时只能纠正 $(d-1)/2$ 个随机比特错误。

图 9-5 给出了包级别 FEC 编、解码过程的示意图。在解码时我们需要知道丢失包的位置以便选择 $\boldsymbol{G}$ 中正确的行构成 $\boldsymbol{G}_k$，如前所述，这是可以通过低层或其他协议获得的。求出 $\boldsymbol{G}_k^{-1}$，则可以重建信息符号组

$$\boldsymbol{x}=\boldsymbol{G}_k^{-1}\cdot\boldsymbol{y}' \tag{9-34}$$

其中 $\boldsymbol{y}'$ 为接收的 k 组码字。

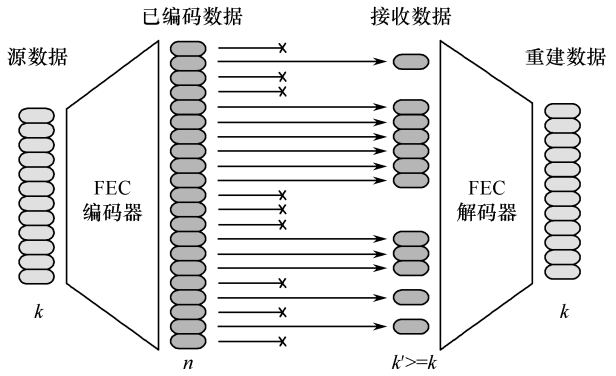


图 9-5　包级别 FEC 的编、解码过程

假设包丢失事件是独立的，我们可以推出在使用包级别的 $(k+1,k)$ 和 (n,k) FEC 后，接收端的丢包率将分别降至 P_1 和 P_{n-k}，如下式（见习题 3）：

$$P_1=p[1-(1-p)^k]$$

$$P_{n-k}=p\left[1-\sum_{j=0}^{n-k-1}\binom{n-1}{j}p^j(1-p)^{n-j-1}\right] \tag{9-35}$$

式中，p 为信道原有的丢包率。

2. 基于 Vandermonde 矩阵和基于 Cauchy 矩阵的 RS 码

包级别的 FEC 通常采用 $q=2$ 扩展域 $GF(2^c)$ 中的 RS 码，因为 $GF(2^c)$ 域中的元素均可用 c 比特表达，这使数据管理很简单；同时在该域上多项式的系数非 0 即 1，加法和减法是对应系数的模 2 和，可以用简单的异或（XOR）操作来实现。

考虑到编解码的方便，RS 系统码被广泛地应用。RS 系统码的生成矩阵 $\boldsymbol{G}$ 由一个 $k\times k$ 的单位矩阵 $\boldsymbol{I}_k$ 和一个 $(n-k)\times k$ 的矩阵 $\boldsymbol{P}$ 构成：

$$\boldsymbol{G}=\begin{bmatrix}\boldsymbol{I}_k\\ \boldsymbol{P}\end{bmatrix} \tag{9-36}$$

式中，$\boldsymbol{P}$ 为 Vandermonde 矩阵。此时生成的码称为基于 Vandermonde 矩阵的 RS 码，也是经典的 RS 码。如前所述，在解码端根据正确接收的编码包的位置，选择 $\boldsymbol{G}$ 中对应的行构成子生成矩阵 $\boldsymbol{G}_k$，通过 $\boldsymbol{G}_k^{-1}$ 就可以恢复出原有的 k 个信息包。

在实际中还常常使用另一种矩阵，称为 Cauchy 矩阵，来作为 $\boldsymbol{P}$，此时生成的码称为基于 Cauchy 矩阵的 RS 码，它是经典 RS 码的一个变种。

设 $t_0,t_1,\cdots,t_s$ 是 $GF(p)$ 中不相等的元素，$r_0,r_1,\cdots,r_t$ 也是 $GF(p)$ 中不相等的元素，且 t_i+r_j

$\neq 0$,$(0 \leqslant i \leqslant s-1, 0 \leqslant j \leqslant t-1)$,则矩阵

$$\begin{bmatrix} \frac{1}{t_0 + r_0} & \frac{1}{t_0 + r_1} & \cdots & \frac{1}{t_0 + r_{t-1}} \\ \frac{1}{t_1 + r_0} & \frac{1}{t_1 + r_1} & \cdots & \frac{1}{t_1 + r_{t-1}} \\ \vdots & \vdots & & \vdots \\ \frac{1}{t_{s-1} + r_0} & \frac{1}{t_{s-1} + r_1} & \cdots & \frac{1}{t_{s-1} + r_{t-1}} \end{bmatrix} \tag{9-37}$$

称为 **Cauchy 矩阵**。Cauchy 矩阵的任何一个 $k \times k$ 子矩阵都是可逆的,而求解 Cauchy 矩阵的逆比一般矩阵容易。因此,基于 Cauchy 矩阵的 RS 码的解码要简单一些。

3. 数据包分段

用于纠正线路比特错误的传统 FEC 通常通过硬件实现,而包级别 FEC 是在传输层之上进行的,一般由软件实现,因此高效的算法就显得十分重要。首先,我们需要考虑如何对几百甚至几千比特长的包进行编解码操作;第二,FEC 涉及有限域内多项式的算术运算,信息数据必须首先转换成域内多项式,然后进行运算并取模后再转换回来,如何有效地进行这些复杂度相当高的计算,是另一个需要考虑的问题。

下面我们先来考虑第一个问题。

图 9-6 是将长数据包(如一帧视频数据)分割后进行包级别 FEC 的示意图。如图 9-6(a)所示,一帧数据被分成 k 段,每段包含 b 个比特。如果最后一段较短,则填充零至 b 比特,然后按纵向完成(n,k)RS 码的校验数据的计算,得到$(n-k)$个校验包(图中灰色的包)。正确解码必须知道所使用的 n,k 值和每个包所在的位置(序列号),因此需要通过(如图所示的)附加 FEC 头或以其他方式将这些信息传送给接收端。IETF 为此已经为 FEC 规定了 RTP 负载的格式(RFC 2733)。

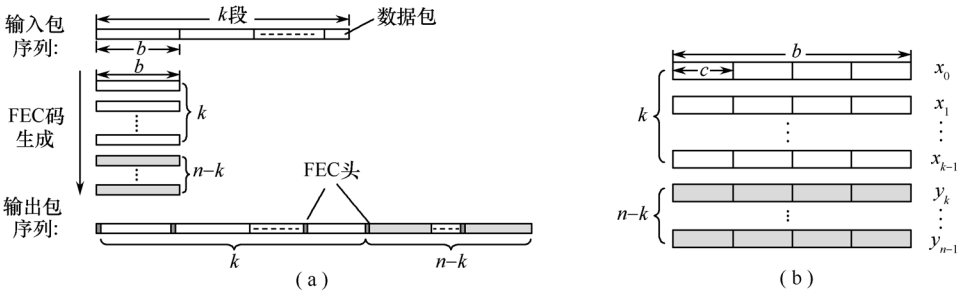


图 9-6　数据包分段

一般说来,b 的大小根据应用的需要而确定。值得注意,由于接收端接收到 k 个段后才能解码,b 大则延时加大;而 b 过小,包头(包括 IP 包头和传输层包头)所占比特数的比例加大,使传输效率降低。

4. 编码与解码算法

研究者已经提出了若干高效的 RS 删除恢复码的编、解码算法[4-6],在这里我们仅介绍其中的一种,它在 Pentium 2.8 GHz 的计算机上可以实现 200 Mb/s 的输出[6]。

假设将数据包中的 b 个二进制符号按 c 比特为一个单元划分,(b 应能被 c 整除)[见图 9-6(b)],每个单元代表一个 2^c 进制的符号,即 GF(2^c)中的一个元素,那么信息数据包 $\boldsymbol{x}_i, i=0,1,\cdots,k-1$,和编码后的数据包 $\boldsymbol{y}_j, j=0,1,\cdots,n-1$,都可以表示成包含 $l=b/c$ 个 GF(2^c)元素的矢量。

选择 c 等于计算机的字长,如 $c=8$、16 或 32,可以提高计算效率。值得注意,c 决定了有限域的符号集大小,也就决定了编码后包的最多个数。例如,$c=8$,对应的有限域为 $GF(2^8)$,其码长为

$n=2^8-1$(见 9.3.2 节有关 BCH 码的叙述),因此,无论信息数据包的个数 k 为多少,编码后的数据包的个数不可能超过 n 这个数值。

下面我们在将信息数据包 $\boldsymbol{x}_i$ 和编码后的数据包 $\boldsymbol{y}_j$ 都看成以 GF(2^c)内元素构成的矢量的情况下,来讨论包级别 RS 编码和解码运算。

当进行 RS 编码时,根据(9-33)式,我们有

$$\boldsymbol{y}_j = \sum_{i=1}^{k} \beta_{ij} \boldsymbol{x}_i \tag{9-38}$$

其中 β_{ij} 为生成矩阵第 j 行第 i 列的元素。这相当于对数据包中 l 个 c 比特符号并行编码。由(9-27)式和(9-29)式可知,生成矩阵的元素是有限域内的非零元素,因此上式包含了 GF(2^c)域上标量与矢量的相乘和相加运算。

当进行 RS 解码时,首先求出 $\boldsymbol{G}_k^{-1}$,然后根据(9-34)式,

$$\boldsymbol{x}_i = \sum_{j=1}^{k} \beta'_{ij} \boldsymbol{y}_j \tag{9-39}$$

式中,β'_{ij} 为 $\boldsymbol{G}_k^{-1}$ 的第 i 行第 j 列的元素,上式也涉及 GF(2^c)域上标量与矢量的相乘和相加运算。

(9-38)式和(9-39)式中的标量与矢量相乘、相加和矩阵求逆 $\boldsymbol{G}_k^{-1}$ 都是在有限域上进行的,下面我们分别讨论如何高效地进行这两种运算。

(1) 有限域上的标量与矢量相乘和相加

我们将(9-38)式和(9-39)式写成一个一般的式子:

$$\boldsymbol{z} = \boldsymbol{z} + \beta \boldsymbol{w} \tag{9-40}$$

上式中用迭代方式来完成求和,迭代的次数由求和的项数决定。将上式改为以矢量的分量来表示,则得到

$$z_m = z_m + \beta w_m \qquad (m = 0,1,\cdots,l) \tag{9-41}$$

其中 z_m,β 和 w_m 都是 GF(2^c)中的元素。

根据有限域的知识我们知道,域内所有元素都可以用本原元 α 的幂来表示,且 α 的幂以(2^c-1)为周期重复,因此(9-41)式可以写为

$$z_m = z_m + exp\ |\log(\beta) + \log(w_m)\ |_{2^c-1} \tag{9-42}$$

其中$|\cdot|_{2^c-1}$表示对 2^c-1 求模。如果所使用的 RS 码的域 GF(2^c)内元素个数不太多,则可以将这些“指数”和“对数”事先制成表,上式的运算则由两次对数查表、一次相加、一次指数查表和一次相加完成。后一次加法是 GF(2^c)域上的相加,只是一个逐比特的异或运算。

表 9-1 以 GF(2^3)域为例给出了在模(x^3+x+1)运算下构成 GF(2^3)的多项式元素及它们的幂指数、对数和数值表示之间的对应关系。表中元素的数值表示方法在 9.3.2 节有关有限域的介绍中曾经讲过,只不过那里给出的是二进制形式。

表 9-1 GF(2^3)元素列表

数值表示	GF 元素	GF 元素的幂指数形式	GF 元素的对数
0	0	0	/
1	1	α^0	0
2	x	α^1	1
4	x^2	α^2	2
3	$x+1$	α^3	3
6	x^2+x	α^4	4
7	x^2+x+1	α^5	5
5	x^2+1	α^6	6

(2) Cauchy 逆矩阵 $\boldsymbol{G}_k^{-1}$ 的求解

假设接收端收到的 k 个编码数据包中有 $k-s$ 个是原来的信息包，为了方便起见，我们将这些包移到前面，它们在生成矩阵中的行索引是$\{r_s, r_{s+1}, r_{k-1}\}$；而接收到的校验包和未被接收到的信息包的索引分别是$\{t_0, t_1, \cdots t_{s-1}\}$和$\{r_0, r_1, \cdots, r_{s-1}\}$，那么接收到的编码信息包构成子生成矩阵 $\boldsymbol{G}_k$：

$$\boldsymbol{G}_k = \begin{bmatrix} \boldsymbol{I}_{k-s} & 0 \\ \boldsymbol{A} & \boldsymbol{B} \end{bmatrix} \tag{9-43}$$

其中

$$\boldsymbol{A} = \begin{bmatrix} \frac{1}{t_0 + r_s}, & \frac{1}{t_0 + r_{s+1}}, & \cdots, & \frac{1}{t_0 + r_{k-1}} \\ \frac{1}{t_1 + r_s}, & \frac{1}{t_1 + r_{s+1}}, & \cdots, & \frac{1}{t_1 + r_{k-1}} \\ \vdots & \vdots & & \vdots \\ \frac{1}{t_{s-1} + r_s}, & \frac{1}{t_{s-1} + r_{s+1}}, & \cdots, & \frac{1}{t_{s-1} + r_{k-1}} \end{bmatrix} \tag{9-44}$$

$$\boldsymbol{B} = \begin{bmatrix} \frac{1}{t_0 + r_0}, & \frac{1}{t_0 + r_1}, & \cdots, & \frac{1}{t_0 + r_{s-1}} \\ \frac{1}{t_1 + r_0}, & \frac{1}{t_1 + r_1}, & \cdots, & \frac{1}{t_1 + r_{s-1}} \\ \vdots & \vdots & & \vdots \\ \frac{1}{t_{s-1} + r_0}, & \frac{1}{t_{s-1} + r_1}, & \cdots, & \frac{1}{t_{s-1} + r_{s-1}} \end{bmatrix} \tag{9-45}$$

$\boldsymbol{G}_k$ 的逆 $\boldsymbol{G}_k^{-1}$ 可以表示为

$$\boldsymbol{G}_k^{-1} = \begin{bmatrix} \boldsymbol{I}_{k-s} & 0 \\ 0 & \boldsymbol{B}^{-1} \end{bmatrix} \begin{bmatrix} \boldsymbol{I}_{k-s} & 0 \\ -\boldsymbol{A} & \boldsymbol{I}_s \end{bmatrix} \tag{9-46}$$

由于 Cauchy 矩阵的性质，任何一个 Cauchy 矩阵 $\boldsymbol{B}$ 的行列式可按下式计算：

$$\det(\boldsymbol{B}) = \frac{\prod_{i<j}(t_i - t_j) \prod_{i<j}(r_i - r_j)}{\prod_{i,j=0}^{s-1}(t_i + r_j)} \tag{9-47}$$

利用公因子，Cauchy 矩阵的逆 $\boldsymbol{B}^{-1}$ 为

$$\boldsymbol{B}^{-1} = [d_{i,j}]_{i,j=0,\cdots,s-1} \tag{9-48}$$

其中

$$d_{ij} = (-1)^{i+j} \frac{e_j f_i}{a_j b_i (t_j + r_i)} \tag{9-49}$$

$$a_m = \{\prod_{i<m}(t_i - t_m)\}\{\prod_{i>m}(t_m - t_i)\} \tag{9-50}$$

$$b_m = \{\prod_{i<m}(r_i - r_m)\}\{\prod_{i>m}(r_m - r_i)\} \tag{9-51}$$

$$e_m = \prod_i (t_m + r_i) \tag{9-52}$$

$$f_m = \prod_i (t_i + r_m) \tag{9-53}$$

在解码时，$\boldsymbol{G}_k^{-1}$ 与接收到的包相乘[见(9-39)式]，根据式(9-42)，只有 $\log(a_m)$、$\log(b_m)$、$\log(e_m)$、$\log(f_m)$和 $\log(d_{ij})$需要计算。

5. 其他的删除复原码

最近 10 年来对删除复原码的研究十分活跃，出现了许多不同类型的删除复原码；同时删除复原码也越来越多地应用于多播协议、大规模存储、数据广播和分布式计算领域。

从删除复原码的构成来说，大致可以分成两类。一是最大距离分离码 **MDS**(Maximum-Distance-Separable Codes)，另一类是**稀疏图码**(Sparse graph Codes)。所谓 MDS 码是指接收到的码长等于信息长度就足以恢复原信息的码。前面介绍的 RS 码就是一种 MDS 码；其他构成 MDS 码的方法还有信息离散算法 **IDA**(Information Dispersal Algorithm)、**阵列码**(Array Codes)等。低

密度奇偶校验码 **LDPC**(Low Density Parity Check Codes)的优良特性引发了另一类删除复原码——稀疏图码的诞生,其中最为著名的一类称为**喷泉(Fountain)码**。初始的喷泉码称为 **Tornado** 码,后来发展为通用删除码(Universal Erasure Codes),或称 **LT 码**。LT 码的进一步扩展成为 **Raptor 码**。对于这类码,在接收端收到 $k+\varepsilon$ 个包后能够以$(1-\delta)$的概率恢复出 k 个信息包来,ε 和 δ 都是比较小的数值。

我们知道由于所使用的有限域的限制,RS 码的 n 和 k 的取值不能超过一定的数值,例如对于 $GF(2^8)$,$k \leqslant n \leqslant 256$。当要传送的数据块很大(如大文件)时,需要将数据分成段,这将降低抵御删除错误的能力。同时,n 和 k 的数值很大时,RS 编码和解码的时间可能长到不可接受的地步。而稀疏图码可以克服上述两个缺点,因而在大文件的可靠传输中得到了广泛的应用,例如应用于 BT 下载、文件传输 FLUTE(File Delivery over Unidirectional Transport)[7] 和可靠多播 RMT(Reliable Multicast Transport)[8] 等协议中。

9.3.4 数据交错

数据交错(Interleave)是提高 FEC 纠正突发错误能力的一种方法。用交错方法构造的码称为**交错码**。形成交错码的方法是,将 i 个分组码排成一个矩阵[见图 9-7(a)],每一行是一个(m,k)分组码,i 称为**交错度**。发送端按列的次序自左至右进行传输;接收端接收到此码流后,仍排成与发送端同样的矩阵[见图 9-7(b)],然后以行为单位按原分组码的译码方式译码。图 9-7(a)和(b)中用箭头分别表示出传输顺序和解码顺序。如果原来的分组码能纠正长度小于和等于 t 的错误,经过交错以后则能纠正所有长度小于和等于 $i \times t$ 的突发错误。

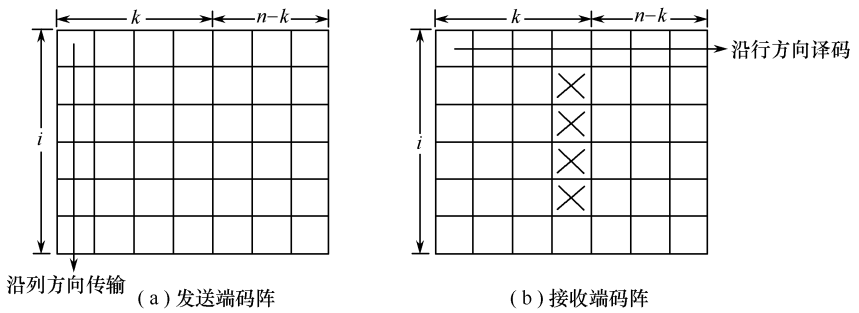


图 9-7 数据的交错

图 9-7(b)中用×表示一个长度为 $m(m<i)$的突发错误。错误连续地发生在码阵的列上,因为在传输过程中数据是按列排列的;而解码时,由于是按行解码,这些错误就被分散到行上,每行只有一个错误,即将突发错误分散成了随机错误,因此,每行的(n,k)分组码可以将各行的随机错误纠正。对于 $m>i$ 的突发错误,只要分散到每一行上的错误少于 t,都能为分组码所纠正。此外,交错方法可以使用较短的分组码,这比具有同等纠错能力的其他码译码简单。由于上述优点,在信道的传输条件不理想时,数据交错技术与 FEC 一起得到广泛的应用。但是必须注意,使用交错码时需要有较大的缓存区,并引入较大的系统延时。

图 9-8 给出了在进行包级别 FEC 时采用数据交错的示意图。图中编码时每三个信息数据包产生一个校验包(横向),然后按列进行传输;在接收端收到 4 组包后,按行进行解码。

9.3.5 前向差错控制与 ARQ 的结合

我们知道,ARQ 引入较大的延时,而 FEC 引入冗余数据降低了带宽利用率,尤其在丢包率较低的情况下更为显著。在一些应用中,将二者结合起来使用可以有效地提高数据传输的可靠性和

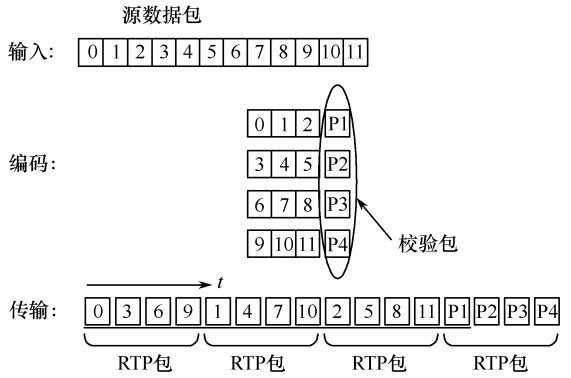


图 9-8　包级别的数据交错

带宽利用率。图 9-9 给出了二者结合的两种形式:分层式和集成式。

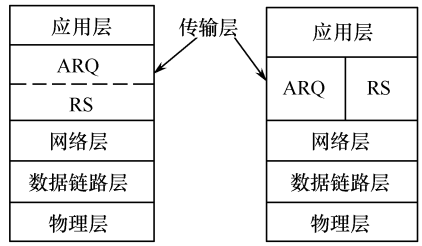


图 9-9　分层式和集成式的混合 FEC 与 ARQ

在**分层式**中,RS 码和 ARQ 是独立操作的。如果 RS 解码器接收到 k 个以上的数据包并正确恢复出 k 个原始信息包,则将解码结果递交给上层;如果接收到的数据包少于 k 个,ARQ 则发出重传请求,直到接收到 k 个包为止。**集成式**的 FEC 和 ARQ 可以有多种形式,例如一种方式是,接收端每检测到丢失一个包就通过 NAK 请求重发一个新的包,直到收到 k 个数据包解码出原来的 k 个原始信息包;发送端则是在依次发送完 k 个信息包后,按重传请求的个数发送校验包。

FEC 和 ARQ 结合使用能够显著提高端到端的吞吐量,所花费的代价是使用了更多的终端资源。一般来说,集成式比分层式的 FEC+ARQ 性能要好一些,但是复杂程度要高。

9.4　编码层差错控制

9.3 节介绍的传输层差错控制技术是一般通用的错误处理技术,它们在原理上并不依赖于所传送的数据的具体特征,因而这些技术可以应用于视频数据的分组传输。除了通用技术以外,针对视频数据的具体特点,在编码层还存在一些专门的差错防护和补偿措施,这便是本节要介绍的内容。

9.4.1　误码和包丢失对已压缩视频信号的影响

视频压缩编码技术的基本出发点是尽量减少数据中携带的冗余信息,以达到降低码率的目的。但是已编码的码流中携带的冗余信息越少,意味着每个比特携带的信息量越多,误码或包丢失对图像质量所造成的损伤也就越大。

在 MPEG 标准中已编码的数据流具有一定的层次结构,每一层次有一个头(见 4.1.7 节)。如果误码发生在图像序列头中,有关图像分辨率、帧率等信息被破坏,导致不能对整个序列正确地解码。为了防止这种情况的发生,通常在序列发送过程中要重复传送序列头信息。GOP 头信息与解码的关联较小,因而 GOP 头内的误码对解码不会产生严重的影响。误码如果发生在图像头内则破坏性较大,例如 Picture_start_code 发生错误,解码器可能找不到图像的起始处,在最坏的情况下,整帧图像将被迫丢弃。如果丢弃的这一帧是下一帧图像的参考帧,那么下一帧的解码也受到影

响。宏块中的几个参数，如 DCT 的直流系数、宏块的运动矢量等，是参照相邻块的对应参数进行 DPCM 编码的，而 DPCM 在每个条(Slice)的起始处复位(即预测值置为零)，因此，如果误码破坏了经 DPCM 编码的参量，那么整个这一条将不能正确地解码，直到正确地检测到下一条的头，解码器才能恢复正常工作。因此条头常称为**再同步标记**。

条内的数据(DCT 系数和运动矢量等)是经过变长编码(VLC)的。由于 VLC 的码长不固定，只根据码表来解码，因此误码可能导致产生码表之外的无效码字而无法解码；更坏的情况是，误码产生了码表之内长度不同的另一个码，从而使错误传播下去。以表 3-1 的编码方式Ⅱ为例，假设原码流为 0100110…第 1 位码应译为 S_1。如果第 1 位误码使序列改变为 1100110…那么前 3 位构成了码表中的有效码字 S_3，解码器无法察觉错误，因而使原序列本来应译成的 $S_1S_2S_1S_3$…却变成了 $S_3S_1S_3$…这种码字边界发生错位的错误称为**失步**。失步会使错误在空间上自左至右传播，一直延续到下一个条(再同步单元)的开始处为止。

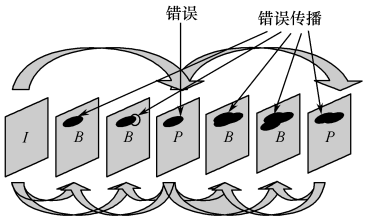


图 9-10　P 帧的错误在时间方向上的传播

此外，通过帧内和帧间预测编码可能导致错误在空间和/或时间上的传播。例如像前面提到的某个宏块的运动矢量或 DCT 的 DC 分量的错误，将导致同一条内的后续宏块的对应参数的错误。又如在 H.264 中可以进行帧内预测编码，参考块中发生的错误会因此而扩散到其他块中。与错误在空间上传播相类似，在参考帧内发生的任何错误通过帧间预测会在时间方向上传播。图 9-10 给出了 P 帧发生的差错向前和向后传播的情况。

最后需要指出，同样的信道错误率对不同码率的信号产生的影响是不同的。以信元丢失率为 10^{-8} 为例，这个数字意味着每 10^8 个信元，或者说 $10^8\times48\times8=3.84\times10^{10}$ 个信息比特，出一次错误。该数字对于 64 kb/s 的数字电话而言，其发生错误的平均间隔时间为 $3.84\times10^{10}/(64\times10^3)\simeq6.9$ 天，而对于码率为 40 Mb/s 的 HDTV，则发生错误的平均时间间隔只有 16 分钟左右。一个电话的通话时间一般以分钟计，因此多数电话不会受到干扰；而一个电视节目一般在 1 小时以上，用户会多次察觉信元丢失的影响。

压缩编码旨在消除信号中的冗余，但是冗余信息的减少将导致误码或包丢失对图像质量的损伤增大。因此，在传输条件不理想(错误率较高)的情况下，可以在进行压缩编码时有意识地保留一些冗余信息，以降低已编码流对差错的敏感性。这类压缩编码方式称为**鲁棒编码**(Robust Coding)，或**抗误码编码**(Error Resilient Coding)。

9.4.2　抗误码编码

由于预测编码和变长编码是导致差错在时间和空间上传播的主要因素，因此本节将首先介绍针对预测编码提出的抗误码措施：泄漏预测、帧内编码刷新和区域受限的帧间预测；然后介绍针对变长编码提出的抗误码熵编码和可逆变长编码。最后我们将在码流的组织方面，介绍编码单元的划分，以及对重要信息的冗余插入、分类和保护，这些措施也可以提高码流抵御错误的鲁棒性。

1. 泄漏预测

在 3.5.1 节关于预测编码的讨论中我们知道，当使用最佳预测器时，预测的均方误差值最小，即预测增益最高。而**泄漏预测**(Leaky Prediction)选择预测器的系数(a_i，$i=1\cdots N$)小于其最佳值 a_{iopt}，即 $0\leqslant a_i\leqslant a_{iopt}$($i=1,\cdots,N$)。这相当于对最佳预测系数乘以一个因子 $L(0\leqslant L\leqslant1)$，称为**泄漏因子**。$L=1$ 表示没有泄漏。$L<1$ 时，泄漏使预测增益和编码效率下降；但是由于预测误差对参考值的依赖性减少，差错沿空间(对帧间预测则为时间)方向上的传播距离随之减小。假设受到错误干扰的帧间预测误差为 D，经过 n 帧之后，D 将衰减到 DL^n。在极端情况下，$L=0$，帧间预测编

码退化为帧内编码，差错在时间方向上完全不会传播。

泄漏因子的大小根据所期望的差错衰减速度和编码效率之间的折中考虑来确定。

2. 帧内编码刷新

减少两个 I 帧之间的预测帧(P 帧或 B 帧)的数目，可以减少差错在时间上传播的范围。但是在要求高编码效率的应用中，不允许在码流中有许多 I 帧，此时可以让连续的帧间编码帧中不同位置上的条(或宏块)轮流采用帧内编码，在一定的时间间隔内完成一帧的循环，差错的传播则被限制在这个时间范围之内。这种方法称为**周期刷新 CIR**(Cyclic Intra Refresh)。例如在 H.261 中，每个 P 帧按扫描顺序依次刷新(帧内编码)一个宏块，对于 CIF 格式的图像，396 帧循环一周。

帧内编码刷新是阻止差错在时间上传播的基本手段。增加每帧刷新的宏块数可以加快差错恢复的时间，但是帧内编码块增加了编码产生的比特数，为了维持码率在信道限制的码率之内，量化步长要相应提高，这意味着在无差错情况下视频质量的降低。因此，如何选择刷新宏块的**数量**和它们的**位置**，以获得差错的鲁棒性和无差错情况下视频质量的最佳折中，是帧内编码刷新中主要考虑的问题。

图 9-11 给出 MPEG4 Annex E **自适应帧内刷新 AIR**(Adaptive Intra Refresh)的示意图。这里每个 VOP(或帧)中刷新宏块的个数是固定的，而其位置选择在图像运动比较剧烈的区域，因为相比于静止区域，这些区域的错误通过帧间预测会扩散到更大的范围中。宏块运动剧烈的程度用该宏块经运动补偿后的帧间预测残差 SAD 来衡量，若 SAD 大于前一个 VOP 所有宏块残差的均值，则认为该宏块属于剧烈活动区域，并在图 9-11 所示**刷新图**(Refresh map)的对应位置上置 1。每帧编码时根据刷新图按扫描顺序依次完成规定块数的帧内刷新(图中为两个黑色块)，例如第 3 帧刷新了刷新图左上角置 1 的两个宏块，第 4 帧则根据刷新图刷新第 2 列的两个宏块，以此类推。与传统的隔若干帧刷新整个 VOP 一次的方法相比，AIR 的输出速率更平稳，而且无差错情况下的视频质量更好。

帧内刷新宏块的数量和位置的选择还可以有许多其他的策略，例如根据网络差错率的大小，动态地选择每帧刷新的块数；每帧在随机的位置上刷新；根据率失真优化原则选择刷新块的位置和数量等。

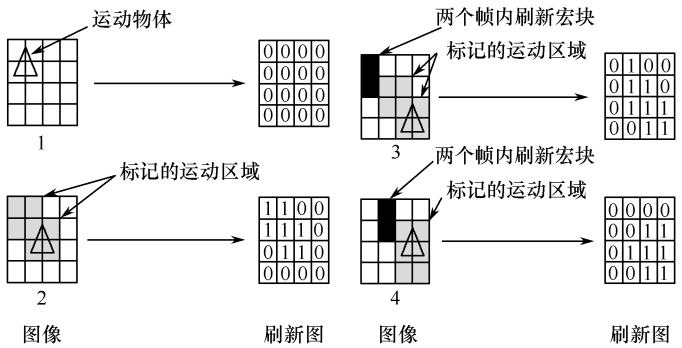


图 9-11　自适应帧内刷新

3. 区域受限的帧间预测

将参考块位置限制在一定的区域之内，能够限制由帧间预测编码引起的差错空间传播的范围。例如，在 H.263 Annex R 中，运动估值只在对应的 GOB(Group of Block，相当于 MPEG2 中的条)中进行，即当前宏块的运动矢量不能指向前一帧同位 GOB 之外的区域。同时，运动矢量在进行 DPCM 编码时也不参考本 GOB 之外的运动矢量。这样，每个 GOB 就是一个独立的可解码单元。如图 9-12 所示，当一个 GOB 丢失(如图中第 2 帧 GOB 3)后，差错只传播到后续帧的对应 GOB(如图中第 3 帧 GOB 3)之内。

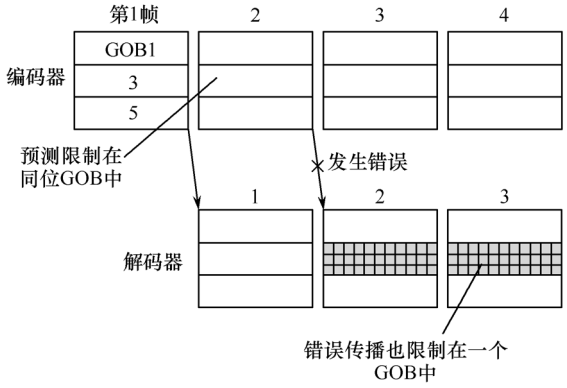


图 9-12　差错在区域受限的帧间编码码流中的传播

4. 抗误码熵编码

误码引起 VLC 解码失步的原因在于，错误地解释了一个码字可能导致错误地识别下一个码字的边界，从而下一个码字的译码也发生错误。如果使用定长码，则一个码字的错误不会干扰对下一个码字的识别。因此**抗误码熵编码 EREC**(Error Resilience Entropy Code)的基本思想就是在保持变长编码高编码效率的同时，在传输时将其放在定长的时段中传送，以避免误码干扰解码器对码边界的识别。

假设有 N 个长度为 b_i 比特的变长码字，$i=1,2,\cdots,N$，编码器首先选择一个能容纳全部数据的时间长度 $T=\sum_i b_i$，并将 T 分成 N 个等长的时隙 $S_i=T/N, i=1,2,\cdots N$；然后利用一个 N 步的算法将 N 个变长码字装入 N 个等长的时隙中。在该算法的第 n 步中，没有被完全装进时隙的第 i 个码字在时隙 $j=i+\phi_n(\mathrm{mod}\ N)$ 中寻找空隙，以便将自己残留的比特部分地或全部地装入 j 时隙，这里 ϕ_n 是一个伪随机的偏置序列。图 9-13 给出上述编码算法的示例。图中 $N=6$，b_i 为{11，9，4，3，9，6}；数据总长 $T=42$，T 被分成 6 个长度 $S_i=7$ 的时隙。假设偏置序列 $\phi_n=\{0,1,2,3,4,5,6\}$。如图 9-13 所示，在第 1 步中，第 3、4、6 个码字完全装进了对应的时隙，而第 1、2、5 个还有部分比特未装进对应的时隙。根据 ϕ_n，码字 1 在时隙 $j=1+1(\mathrm{mod}\ 6)=2$ 中寻找空隙；同理码字 2 和 5 分别在时隙 3 和 6 中寻找空隙。结果码字 2 和 5 分别将 2 和 1 个比特装进了时隙 3 和 6。在第 2 步中，码字 1 和 5 根据 $\phi_n=2$ 分别在时隙 3 和 1 中寻找空隙，并将残留比特放进找到的空隙中。以此类推，经过 6 步，所有码字都放进 6 个时隙中。在解码端以类似的 6 个步骤进行译码。首先从每个时隙的开始，按照码表分别搜索合法的码字，遇到合法的码字则解码，剩余的比特留在该时隙中；如果时隙中的所有比特构不成合法码字，则在后续步骤中根据 ϕ_n 到时隙 j 中寻找后续的比特，直至可以译码为止。

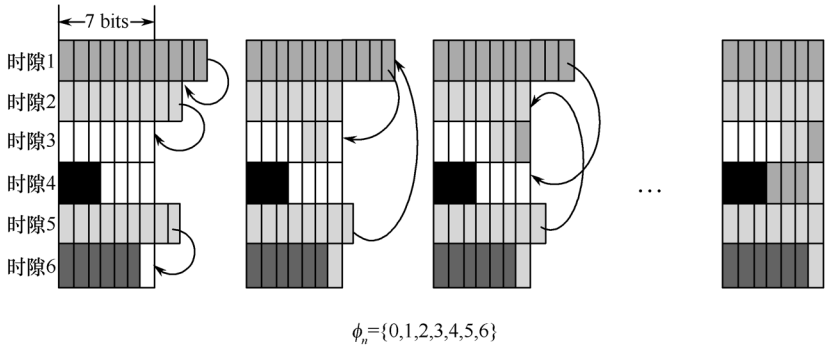


图 9-13　抗误码熵编码举例

在有传输错误时，由于时隙是等长的，每个码字在固定的位置上开始，因此不会造成失步。同时，这种数据排放方法使错误传播造成的译码错误通常发生在分放于多个时隙的长码上，而根据变长编码原理可知，长码对应于小概率事件，因此对主观视觉质量影响较小。

5. 可逆变长编码

如前所述，传输错误可能导致 VLC 解码的失步，因此当发现错误后，不管后面的数据是否正确接收，通常都需将到下一个再同步标记（如条头）之前的数据全部丢弃。为了尽可能地保留可用的数据，**可逆变长编码**（Reversible VLC，RVLC）在正向解码的同时，还允许从下一个再同步标记开始反向解码，当遇到错误时仅把中间部分的数据丢弃。图 9-14 标示出从两个方向解码可能检测到错误（即遇到无效码字）的 4 种不同情况，其中×表示检测到错误的位置，阴影部分为丢弃的数据。

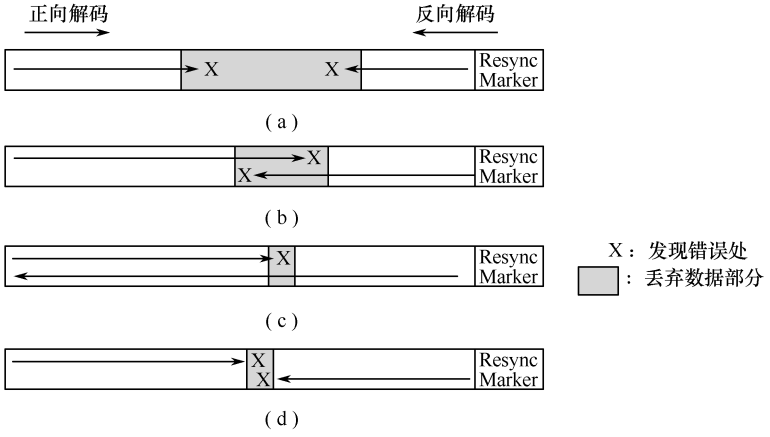


图 9-14　确定双向解码中丢弃数据段的方法

构成 RVLC 码的一种简单的方法是：

(1) 选择一组具有相同**汉明权重**的变长码，即每个码有相同个数的 1（不一定具有前缀特性）；

(2) 在每个码字的前后加上相同的前缀和后缀；

(3) 加上 0 作为最短码。

图 9-15 给出了一个示例。图 9-15(a)给出码表，其中每个 VLC 码字的汉明权重均为 1，加上的前缀和后缀分别为一个 1，则每个 RVLC 码字都含有三个 1。无论从前向还是从后向进行 RVLC 译码，除了最短码是以 0 开头的外，其他码字只要收到三个 1 就表示码字的终结。图 9-15(b)给出了相应的译码过程，图中用 X 表示出现错误的比特，并标出了丢弃的数据段。

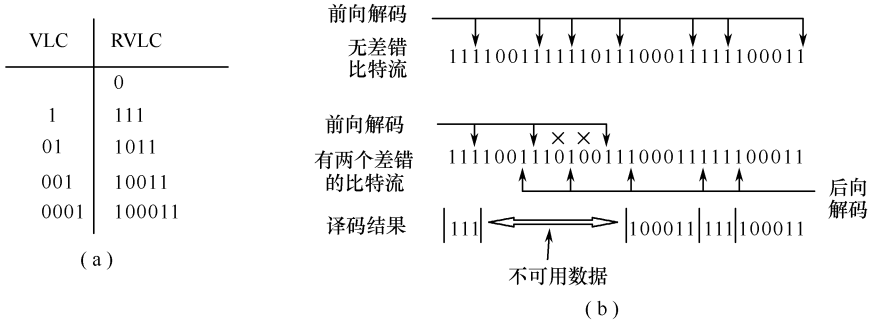


图 9-15　(a)RVLC 码表；(b)传输错误对译码的影响

更复杂的 RVLC 可以由增加更长的前、后缀、或增加比特翻转（0 变 1、1 变 0）的码字构成。除了基于具有相同权重的 VLC 码来构造 RVLC 码外，RVLC 也可以根据其他策略来构成，例如每个 RVLC 码字含有固定数目的 0 和 1（01，10，0011，1100 …）等。一般的 RVLC 与最佳的霍夫曼码相

比大约增加了2%～3%的比特率，但是如果仔细地设计让RVLC与视频训练序列的DCT系数的统计特性尽可能地匹配，则能够使RVLC在保持抗误码能力的同时有好的压缩性能。此外，基于Golomb-Rice码和Exp-Golomb码也可以生成双向可解的码，并保持其良好的编码效率。

在H.263 Annex D和Annex V中，头信息和运动矢量的预测误差采用RVLC进行编码；在MPEG 4中则采用RVLC对纹理信息(DCT系数)进行编码。

6. 再同步单元的划分

如前所述，传输错误可能通过VLC或MV和DC系数的预测编码向后传播，一直延续到下一个再同步单元的开始，因此在发现传输错误时，解码器通常将整个当前再同步单元的数据丢弃。为了使这种情况下图像的损伤尽可能地小，有必要仔细地考虑再同步单元的划分问题。

(1) 减少再同步标记之间的间隔

我们知道，条是MPEG1/2和H.264进行再同步的单元。在H.261/263和MPEG 4中，与条对应的再同步单元分别是GOB和Video Packet。当信道的差错率较高时，可以增加每帧图像的条(再同步单元)的个数，即减少条中所包含的宏块数，以使差错在空间上传播的面积减小。但是条数增多将增加条的头信息的开销，降低编码效率。例如对于4 Mb/s的CCIR 601图像而言，从每条包含44个宏块降至4个宏块，编码效率降低12%左右；在无差错发生的情况下，编码效率的下降将导致图像质量下降1 dB。但是如果信道的信元丢失率为10^{-2}，这种方法大约可以得到1 dB至5 dB的图像信噪比增益[9]。

(2) 按相同比特数划分再同步单元

在一些标准例如H.261/H.263中，将同一行或相邻行上的宏块组成一个再同步单元，称为GOB。每个单元内的宏块个数是相同的，且再同步标记(GOB头)都位于图像左侧的起始位置。由于不同区域内的图像复杂程度可能不同，因此编码后每个GOB所包含的比特数不同。图9-16(a)中的黑色矩形块给出了再同步标记在H.263比特流中的位置的示例，再同步标记之间距离较远的部分对应于图像较复杂的区域。在误码率一定的情况下，包含比特数越多的GOB越容易受到传输错误的干扰而被丢弃；而图像复杂区域内的错误在时间和空间上的传播要比在图像简单区域发生的错误更为恶劣。为了对此加以改进，MPEG 4给出了按相同比特数划分再同步单元的选项。在每个再同步单元(MPEG 4 Video packet)中宏块的个数可以不同，但包含大体相同的比特数(见图9-16(b))，这样在图像复杂的重要区域内分布有较密(指在图像空间中)的再同步标记，一旦发生错误能够及早地恢复。在后来的H.263 Annex K中也增加了类似的策略。

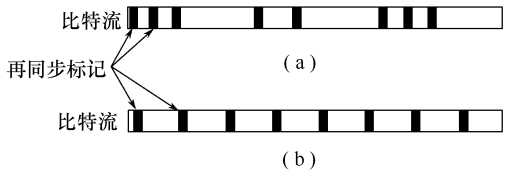


图9-16 再同步标记在码流中的位置：(a)H.263 baseline；(b)MPEG 4

(3) 宏块的灵活排列

如果按扫描顺序将相邻宏块组成再同步单元，当发现传输错误时则意味着相邻宏块将全部丢失。我们将在9.4.3节中看到，这不利于利用空域相关性来进行错误掩盖。为了克服这一缺点，H.264给出了**灵活宏块排列FMO**(Flexible MB Ordering)的选项。在FMO中，定义了条组的概念。**条组**(Slice Group)由一个或若干条(Slice)组成，条由按扫描顺序排列的宏块组成。图9-17所示为两个例子，其中的数字标明组成该条组的条的位置，由图看出，在(a)中条组由交错行上的宏块(条)组成，而(b)中条组由棋盘格位置上的宏块组成。假设(b)中的第2条组丢失，由周围的第1、3、4条组的宏块进行双线性内插则可比较容易地将丢失的宏块近似地恢复出来。

7. 插入或重复重要信息

在已编码流中插入或重复传送某些重要的信息，可以增加对码流进行差错处理的鲁棒性。这种

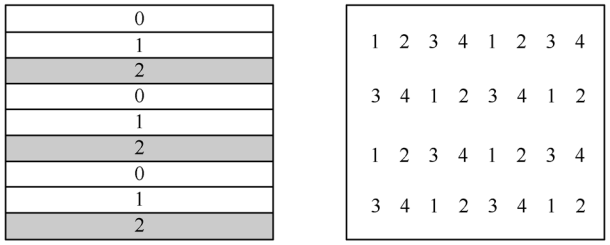


图 9-17　灵活的宏块排列:(a)交错式;(b)棋盘格式

技术有时需要和 9.4.3 节将要介绍的错误掩盖技术结合使用。例如在 MPEG 2 标准中，允许传送帧内编码宏块所对应的运动矢量。我们知道帧内编码的宏块在解码时并不需要运动矢量,在无差错的情况下,该矢量在解码时丢弃;但是如果在传输中发生差错,则可以用该矢量进行错误掩盖(见 9.4.3 节),使差错对主观视觉的影响降低。又如在 H.264 中允许重复传送一帧图像或它的一部分(条),在一般情况下,解码器丢弃冗余的帧或条;而当有传输错误时,则可利用冗余数据来填补受损伤的区域。

我们知道解码所需要的一些最重要的信息,例如图像尺寸、编码类型等通常包含在帧头中,如果帧头由于传输错误而损坏,解码器将由于无法解码而丢弃整个帧。为了降低帧头对错误的敏感性,MPEG 4 中引入了**头扩展码 HEC**(Header Extension Code)。如图 9-18 所示,在每一个 Video packet(再同步单元)中有一个比特的 HEC 域,如果该比特置 1,则它后面跟着有一个重复的帧头(VOP 头)。解码器通过将此帧头与帧起始位置上的帧头进行比较可以确认帧头的正确性;如果帧起始位置上的帧头发生错误,解码器可以利用后续的帧头进行解码而不至于将整个帧丢弃。H.263 Annex W 也有类似的帧头重复措施。

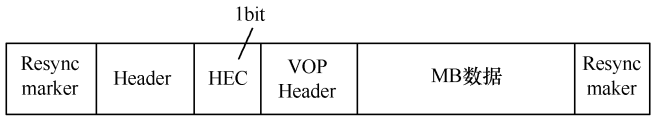


图 9-18　MPEG 4 的头扩展码

8. 数据分割

所谓**数据分割**是指打乱传统的按扫描顺序排列数据的方式,而是将数据按重要性不同进行组合形成码流,以便在后续处理中提高抵御差错的能力。

图 9-19(a)给出了一个 MPEG 4 再同步单元中各宏块数据通常的排列顺序,其中下标表示宏块序列号,COD 表示该宏块是否编码,MCBPC 表示宏块的编码模式和色差块是否编码,CBPY 表示宏块中的 4 个块是否编码,DQUANT 表示当前宏块与前一个宏块量化参数的增量,后面是经差分编码之后的 MV 和 64 个 DCT 系数值。上述信息是按宏块排列的,且 MV 和 DCT 系数是变长编码的,因此解码器检测到错误时无法定位发生错误的具体位置,只能将整个 Video Packet 丢弃。图 9-19(b)给出了使用 MPEG 4 数据分割时各个数据分量的排列位置,在这里所有宏块与运动有关的信息(当编码基于对象时还包括对象的形状)和与 DCT 有关的信息用一个称为 **MBM**(Motion/DC Boundary Mark)的字段分开。MBM 是一个特殊选择的码字,它与 MV VLC 表中任何可能的有效码字组合均保持汉明距离为 1,因此能够从 MV VLC 表中唯一地解码。由于 MBM 的存在,解码器通过是否检测到 MBM 可以判断错误发生在运动部分还是纹理(DCT)部分。如果错误发生在运动部分,整个 Video Packet 被丢弃(或置为 Skip 模式);如果发生在纹理部分,则只丢弃纹理部分,正确接收的运动信息仍可以用来进行运动补偿,从而能够恢复出质量稍差的图像。在当前再同步单元中没有检测到错误但没有发现下一个再同步标记时,也无需丢弃整个的当前单元,

只需丢弃后面的纹理部分，因为 MBM 已被正确地检测到，从而保证运动部分是正确接收了的。

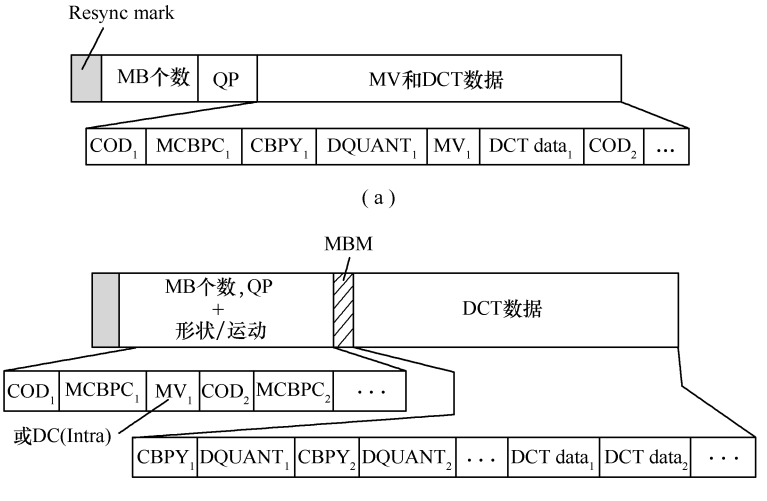


图 9-19　MPEG 4 Video Packet 中各数据分量的位置
(a)无数据分割；(b)采用数据分割

H.264 的数据分割是另一个典型的例子。为了便于与不同传输层协议(如 RTP/IP、H.32x 和 MPEG 2 系统层等)的配合，H.264 在编码层之下增加了一个**网络抽象层**(Network Abstraction Layer，NAL)，如图 9-20 所示。编码后的数据先进行数据分割，然后在 NAL 层被打成 NAL 包再送到传输层。每一个 NAL 包是一个具有整数个字节的逻辑数据单元，NAL 的包头指示该包内数据的类型，其载荷包含实际的数据。

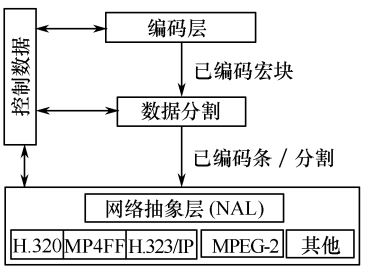


图 9-20　H.264 NAL 层

在 H.264 中，将一段时间内不改变而影响多个帧解码的重要数据(如序列头和帧头)集中在一起构成“**参数集**”。其中**序列参数集**包括影响若干帧解码的参数；**图像参数集**包括影响一帧图像解码的参数，如序号、条组数量和熵编码方式等。由于其重要性，这些参数集可以以非常灵活的方式，例如通过具有高传输可靠性的手段传送(详见下面的不均衡保护)。而一个条内的数据可以放在一个 NAL 包中[见图 9-21(a)]，也可以分割成 A、B 和 C 3 部分[见图 9-21(b)]，其中 A 包含条头、各宏块的头以及运动矢量等，B 包含各宏块帧内预测的残差，C 则包含帧间预测的残差。被分割成 A、B 和 C 3 部分的数据分装在 3 个 NAL 包中，NAL 头中的类型说明该包的数据属于哪一部分。未经数据分割的 NAL 包的丢失造成整个条的丢失；而经数据分割之后，只有在传输中 A 的数据被破坏，整个条才丢弃；如果能够收到包含 A 的 NAL 包，则可以解码 A 和 B(C 被破坏)或解码 A 和 C(B 被破坏)得到质量降低的图像。

MPEG 2 也具有类似的选项，它将数据按重要性分成两部分，分别称为 Partition 0 和 Partition 1。表 9-2 列出了它的 4 种数据分割方式。

表 9-2　MPEG 2 数据分割方法

Partition 0	Partition 1
1. 序列头、GoP 头、图像头和条头	其余数据
2. 以上各项，再加宏块头	其余数据
3. 以上各项，再加运动矢量	其余数据
4. 以上各项，再加 n 个已编码的 DCT 系数	其余数据

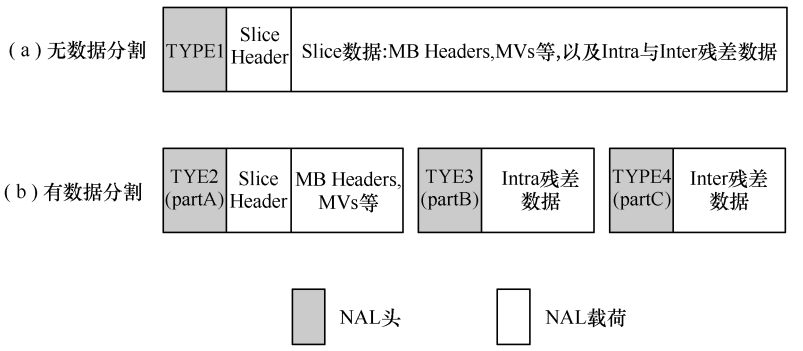


图 9-21　一个条的 NAL 单元

9. 不均衡保护

当数据分割成比较重要和不太重要的部分之后，对它们分别加以不同程度的保护措施，或采用可靠性不同的逻辑子信道传输，称为**不均衡保护**。数据分割加上不均衡保护能够有效地提高编码的容错性能。对不同重要性信息的不均衡保护可以通过多种手段来实现。例如在支持不同优先级传输的网络中，如 IPv6 或 Diffserv 等，重要的信息采用高优先级传输、不重要信息采用低优级传输；或在无线网络中，重要信息采用高功率、不重要的采用低功率传输。又如，对重要信息加以强的 FEC 或长的交错深度保护，非重要信息以弱的、甚至不加 FEC 或数据交错保护等。采用不同的传输协议是实现不均衡保护的另一种方法，例如视频数据本身采用 RTP/UDP 传输，特别重要又不变动的数据，如 8.6.2 节中的元数据、H.264 的参数集等可以采用更可靠的传输机制（如 TCP 或 H.245）传输。当这类信息采用与视频数据不同的逻辑子信道（协议）传输时，称为**“带外”**传输。

9.4.3　错误掩盖

我们在 9.3 节中讲过，FEC 和数据交错只能在一定程度上纠正传输错误；无限制地自动重传虽然可以保证数据传输的可靠性，但在实时性要求较高的多媒体应用中又不宜采用，因此，接收端接收到的码流仍然可能存在传输错误。9.4.2 节中讲述的抗误码编码能够在一定程度上抑制错误的影响，但是传输错误还是会导致解码后图像质量的损伤。所谓**错误掩盖**（Error Concealment）就是在解码端采取一定措施部分地恢复被传输错误所破坏的图像，使其对人的视觉的影响减至最小。不难理解，恢复被破坏图像的基本思路是利用视频图像在空间和时间上的相关性；同时，人的视觉系统可以容忍一定程度的信号失真，这些就构成了实现各种错误掩盖算法的基础。

1. 错误检测与定位

在进行错误掩盖之前必须首先进行错误的**检测**和**定位**，也就是说，必须首先知道在图像的什么位置和什么范围内发生了错误。在 9.3.3 节我们知道在删除信道上，网络的高层（应用层和传输层）或者收到正确无误的包，或者包丢失；而且包丢失的位置可以通过低层协议或其他协议（如 RTP 序列号）准确地获知。如果在发送端根据应用层分帧的思想来封装传输层及底层的包，例如将可独立解码的再同步单元、或数据分割的一个部分、或 FEC 的一个段封装成一个传输层包，那么已知丢包的位置就可以推断出图像受损伤的对应位置。

基于 IP 的网络一般认为属于删除信道，但是 IP 网络的底层（链路层和物理层）可以有许多不同的类别并遵从各种不同的协议，例如有线网或无线网、3G 或 WLAN 等，每一个具体的网络能否视为一个完全的删除信道，取决于其物理信道错误的性质（随机比特错误、突发错误或节点缓存器溢出等）以及物理层和链路层的检错与纠错措施（是否包含对包载荷的检错与纠错、检错与纠错的强度等）。在这里**完全的删除信道**是指，所有的错误在底层都被检测出，且只要有不能纠正的错误

就将包丢弃，因而在传输层包中不含有任何比特错误。不满足这个条件的IP网络我们称为不完全的删除信道。在不完全删除信道上或直接构建在面向比特的信道(如有线或无线的电话网)上的多媒体系统，其解码器的输入码流中可能包含比特错误，此时就需要在解码层进行错误检测和定位了。

解码层错误检测和定位的主要依据是码流句法的合法性和语义的合理性，下面列举了一些典型的判则：

(1) 码字的合法性：例如是否是VLC表中合法的码字，DCT系数值是否在允许的取值范围之内等；

(2) 数据单元数目的合法性：例如8×8块的DCT系数个数是否超过64，宏块中的块数是否超过编码模式给出的数值，宏块数是否超出条头中给出的数值等；

(3) 参数取值的合理性：例如解码后的MV值是否超出给定的运动估值搜索范围，量化参数是否在标准规定的范围之内等。

当根据上述判则在码流中检测到问题时，我们并不能确定传输错误就发生在检测到问题的地方，也有可能发生在此前的某处(见图9-22)。因此在检测到问题时，通常的做法是将当前再同步标记到下一个再同步标记之间的数据全部丢弃(除了在采用RVLC或者数据分割时可以挽回一部分数据)。

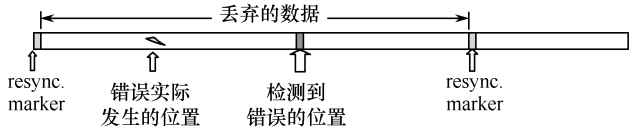


图9-22　差错检测与定位

2. 空间域错误掩盖方法

自然图像中相邻区域亮度和色彩的低频变化是平缓的，因此如果一个或几个MB丢失，最简单的掩盖方法就是利用周围正确接收的宏块内插出丢失的块来。图9-23给出了一个例子，图中粗线框表示丢失的块。如图9-23所示，通常只采用最靠近丢失块的边缘像素来进行内插，内插得到的丢失块中位于(x,y)的像素值$p(x,y)$为

$$p(x,y)=\frac{1}{d_{\mathrm{L}}+d_{\mathrm{R}}+d_{\mathrm{T}}+d_{\mathrm{B}}}\left[d_{\mathrm{L}}p_{\mathrm{R}}(x_2,y)+d_{\mathrm{R}}p_{\mathrm{L}}(x_1,y)+d_{\mathrm{B}}p_{\mathrm{T}}(x,y_1)+d_{\mathrm{T}}p_{\mathrm{B}}(x,y_2)\right] \tag{9-54}$$

式中，d_{L}、d_{R}、d_{T}、d_{B}分别为丢失像素到左、右、上和下边缘像素的距离。为了避免使用距离丢失像素过远(相关性较小)的边缘像素，也可以将丢失块分为4部分(如图虚线所示)，每一部分内的像素只用距自己最近的两个边缘像素来内插，例如图中用$p_{\mathrm{R}}(x_2,y)$和$p_{\mathrm{B}}(x,y_2)$来估计$p(x,y)$。

在空间域内进行错误掩盖的另一种简单方法是，用周围正确接收块的亮度和色度均值(或中值)来代替丢失的块。这可以直接在像素域内进行计算，也可以利用周围块DCT域的直流分量(或加上低频分量)的均值(或中值)作为丢失块的对应DCT系数值来实现。

以上两种方法是基于灰度平滑变化的思想的，因此所得到的掩盖块会模糊原有图像的细节。利用边缘连续性来设计算法可以在一定程度上改善这种情况。例如图9-24所示，如果在相邻正确接收的宏块中检测到一条延伸到丢失块的明显边缘，那么假设该边缘在丢失块中是一条直线(如图中00′所示)，直线两侧的像素分别采用对应侧邻域正确接收的像素进行内插，则可以在丢失块中保留该边缘，从而在一定程度上改善错误掩盖后图像的主观质量。

在图9-25(a)和(b)中我们分别给出孤立块丢失和一行块丢失时的图像；(c)和(d)分别给出使用双线性内插掩盖后的结果。图9-25(a)和(b)可以看成是使用图9-17所示H.264的两种FMO时，一

个条组丢失所产生的错误图形。由于图像受损失的区域比较孤立，因此有利于进行空间域的错误掩盖。

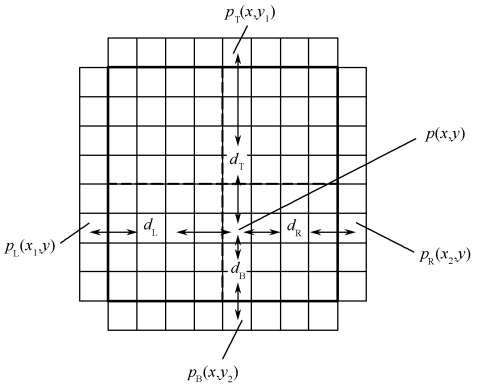


图 9-23　空间域内的错误掩盖

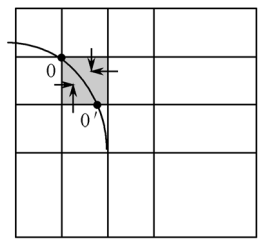


图 9-24　考虑边缘连续性的空域错误掩盖

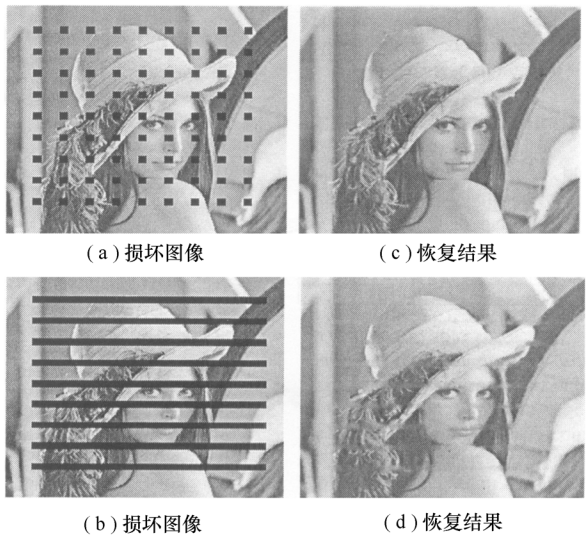


图 9-25　(a)与(b)：受损伤的图像；(c)与(d)：经空间域错误掩盖后的对应图像

3. 时间域错误掩盖方法

利用视频序列在时间上的相关性是进行错误掩盖的另一个途径。当一幅图像有大面积损伤时，利用时域掩盖方法比利用本图像内正确接收部分进行的空域掩盖更为有效。假设图像受损宏块的运动矢量没有丢失、只丢失了纹理(运动补偿后的残差)部分，这种情况通常在编码端采用数据分割和不均衡保护的情况下发生，此时简单地将 MV 指向的前帧参考块补贴到受损宏块的位置上即可。因为只丢失了运动补偿后的残差，错误掩盖的效果是较好的。

当受损宏块的运动矢量丢失或运动矢量与残差都丢失时，首先需要对受损宏块的 MV 进行估计，然后采用 MV 的估计值所指向的参考块来掩盖丢失块。对丢失的 MV 进行估计的最简单的办法是，假设 MV＝0，即将前帧同位宏块复制到丢失宏块的位置。当图像运动较为剧烈时，此种方法的效果将不能令人满意。估计丢失宏块 MV 的其他方法还包括：①令 MV 等于前帧同位宏块的运动矢量[见图 9-26(a)]；②令 MV 等于空间邻域宏块运动矢量的均值或中值[见图 9-26(b)]等。通常一个宏块受损，它的左右邻域的宏块也往往受损，此时可以利用受损宏块上、下相邻宏块的运动

矢量或上下相邻宏块运动矢量的中值或均值来估计受损块的 MV。

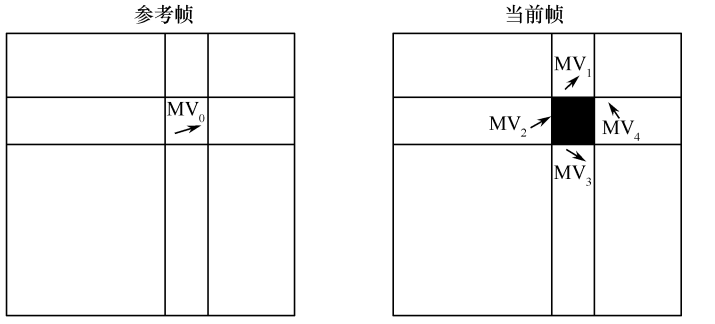


图 9-26　运动矢量的估计

边界匹配算法 BMA(Boundary Matching Algorithm)根据边界匹配情况在一组待选的 MV 中选择一个最佳的估值作为丢失块的运动矢量。待选的 MV 可以包括上述各种方法所估计的 MV 值;所谓边界匹配则是指,用某个 MV 所指向的参考帧数据块来代替丢失块时,替代块的内边缘像素[图 9-27(a)中的深色部分]与丢失块周围宏块边缘像素[(b)图中灰色部分]的差值最小。图9-28给出了在宏块丢失率为 30%时,分别用 MV=0、前帧同位宏块 MV、周围宏块 MV 均值以及 BMA 进行时域错误掩盖的结果。从图(e)可以看到,使用 BMA 算法时,图像的主观质量最好,图像上部的背景有比较整齐的边沿。

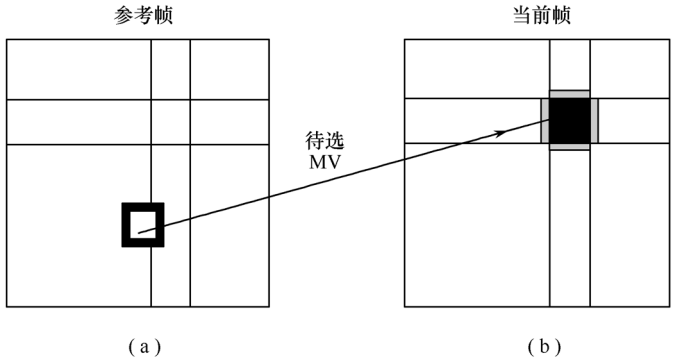


图 9-27　边界匹配算法

不难理解,采用 H.264 FMO 模式使图像受损区域相对孤立,不仅有利于空域错误掩盖,对时域的错误掩盖也是有益的。除了以上介绍的各种比较简单的算法外,无论是对于时域还是空域错误掩盖,研究者还提出了众多的其他算法,性能好的算法通常复杂度较高。在实际应用中,需要在掩盖效果与计算复杂度和实时性之间折中选择。

4. 编码模式的恢复

在有些情况下,特别是没有采用数据分割和不均衡保护时,宏块的编码模式信息可能丢失。此时,恢复丢失宏块编码模式的最简单的办法是,假设该宏块是帧内编码宏块,然后采用空域方法进行错误掩盖。

恢复丢失宏块编码模式的另一种方法是,从周围宏块的编码模式信息来对丢失块的编码模式进行估计。图 9-29 列举了针对 P 帧和 B 帧的一种估计方法,例如图 9-29(a)所示,若丢失块的上、下两个宏块都是前向预测的,丢失块的模式也认为是前向预测的;若其中一个是帧内模式、另一个

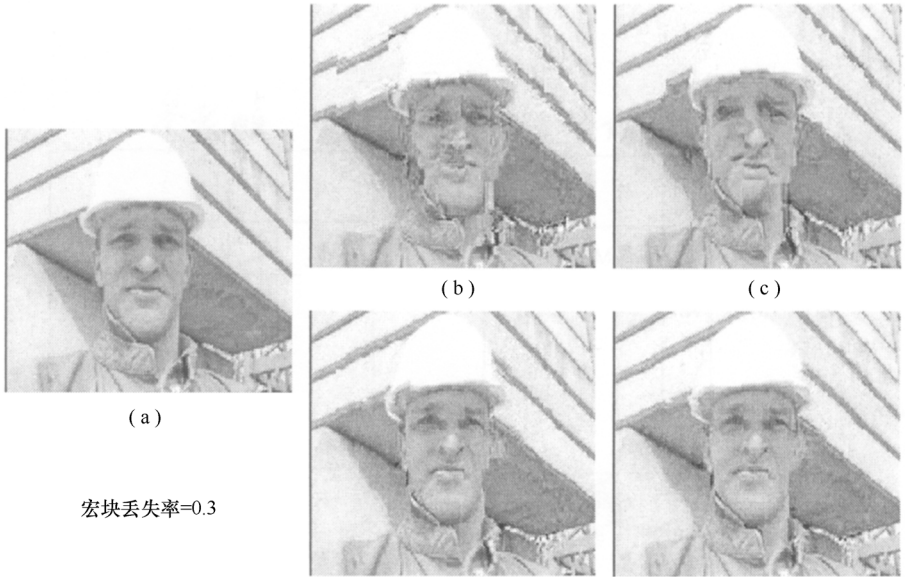


图 9-28　宏块丢失率为 30％时的时域错误掩盖

(a)原始图像；(b)MV＝0；(c)MV＝前帧同位宏块 MV；(d)MV＝周围宏块 MV 均值；(e)BMA 方法

是前向预测模式，则丢失块认为是前向预测模式，等等。

上方MB

下方MB			
	MB 模式	Forw	Intra
	Forw	Forw	Forw
	Intra	Forw	Intra

(a) *P*帧

上方MB

下方MB	MB 模式	Forw	Back	Inter	Intra
	Forw	Forw	Inter	Inter	Forw
	Back	Inter	Back	Inter	Back
	Inter	Inter	Inter	Inter	Inter
	Intra	Forw	Back	Inter	Intra

(b) *B*帧

图 9-29　编码模式的估计

9.4.4　编解码器交互差错控制

前两节分别介绍了编码器和解码器各自独立地采取的差错控制方法，如果在多媒体系统中存在从接收端到发送端的反向信道，则双方可以交互地工作获得更好的差错控制性能。通常这些控制方法基于如下假设：只要差错存在的时间不太长，人是能够容忍的，因此限制差错传播的时间就成为了这些方法的基本出发点。

1. 错误跟踪

在 H.263 使用的这种方法中，编码器保存了最近传输的几帧中的 GOB 错误预测信息；所谓错误预测信息是指第 n 帧的某个 GOB 受损时，第 $n+1$ 到 $n+d-1$ 帧图像中的哪些部分受到影响。如果解码器检测到一个 GOB 受损，它向编码器返回一个否定确认(NAK)信号，如图 9-30所示，解码器返回 NAK 告知第 1 帧第 3 个 GOB 丢失；编码器则根据自己保存的有关第 1 帧第 3 个 GOB

的错误预测信息，对后续帧中可能受到影响的宏块进行帧内编码。在图示的例子中，深色块表示因错误传播而受损的宏块，白色小方块表示帧内编码宏块。由于编码器在第 3 帧收到 NAK，从第 4 帧开始执行帧内编码，因此在接收端错误的传播中止于第 3 帧。

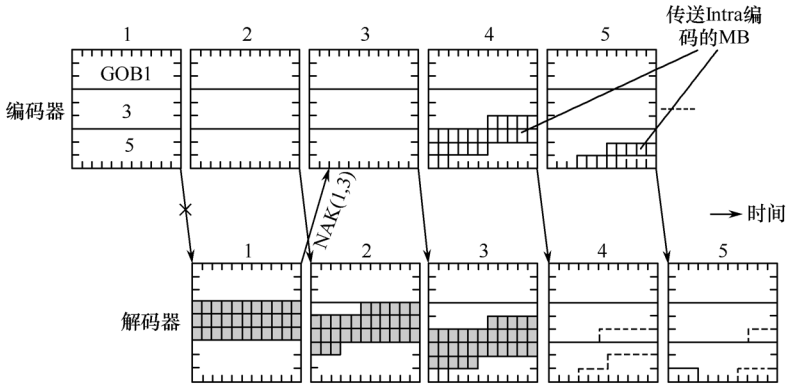


图 9-30　H.263 错误跟踪机制

上述方法的缺点是：①错误跟踪是一个计算量和存储量都很大的过程；②传送大量的帧内编码宏块使码率显著增加。对后者的一个改进方案是，编码端对可能受损的宏块仍进行帧间编码，但将预测时的参考块限制在前帧未受损的区域。

2. 分段独立解码

在 H.263 Annex R 采用的**分段独立解码 ISD**(Independent Segment Decoding)中，运动估值和补偿都限制在特定的区域(如一个 GOB 内)内进行；因此一个 GOB 受损，后续帧中只有对应的 GOB 受到影响，错误不会在空间上扩散。图 9-31 给出了一个例子，从接收端返回的 NAK(2,3)表示第 2 帧第 3 个 GOB 受损。与图 9-30 所示的工作过程类似，编码器将第 5 帧第 3 个 GOB 改为帧内编码。本例的不同之处在于，错误总是限制在各帧的对应 GOB 之内。由于省去了错误跟踪的过程，因此计算复杂度下降；但由于运动估值和搜索范围受限，编码效率有所降低。

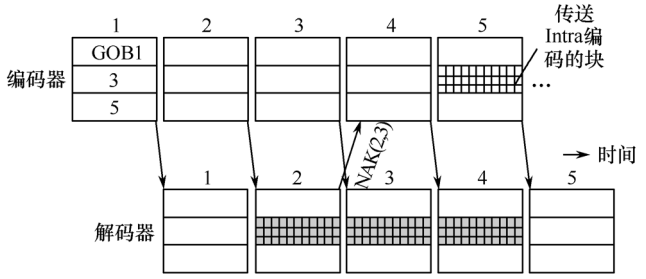


图 9-31　分段独立编码

3. 参考图像选择

在 H.263 Annex N 和 Annex U 采用的**参考图像选择 RPS**(Reference Picture Selection)中，编、解码器都保存着前几帧的解码图像，通过选择正确接收的帧作为参考帧来中止错误在时间上的传播。图 9-32(a)和(b)分别给出了它的两种模式。在图(a)所示的 NAK 模式中，解码器在检测到第 2 帧第 3 个 GOB 丢失时，向编码器返回 NAK(2,3)；编码器得知第 2 帧受损后，则以第 1 帧而不是第 4 帧作为第 5 帧编码的参考帧，并将此信息告知解码端，因此自第 5 帧之后由第 2 帧引起的错误传播中止。由图看出，该方法可以将错误传播的范围限制在信道的双程延时之内。RPS 可以如上所述在帧的级别上进行，也可以在 GOB 或宏块级别上实行，即仅改变受损 GOB 或宏块的参考

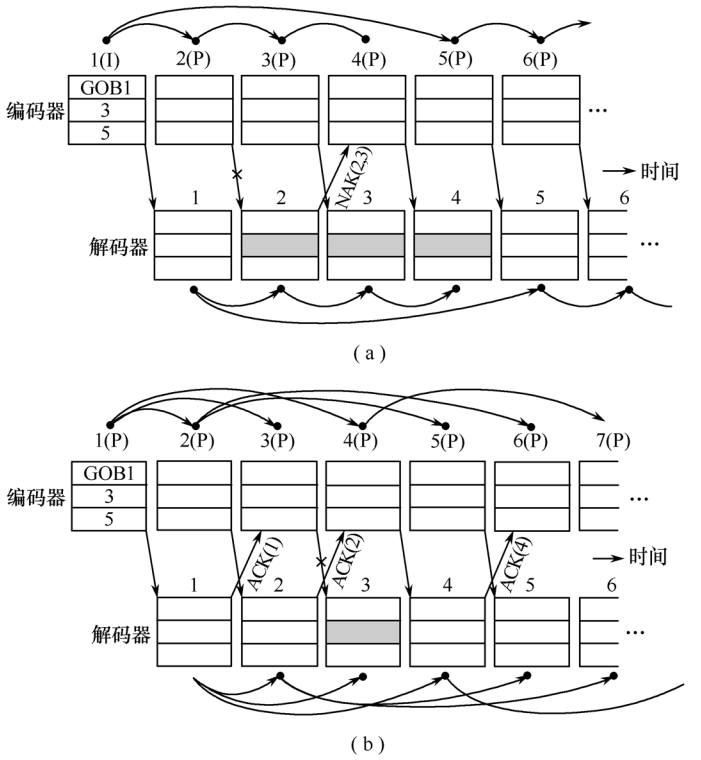


图 9-32　参考图像选择(a)NAK 模式;(b)ACK 模式

图像,其余未受损 GOB 的参考图像仍为距离最近的前一帧,这样可以降低 RPS 在编码效率上的损失。

RPS 的另一种模式如图 9-32(b)所示。在该模式中,解码器对每一个正确接收的帧都会向编码器返回一个 ACK 确认消息,只有被确认过的帧才会作为参考帧。如图中第 3 帧未被确认,因此第 5 和第 6 帧都以第 2 帧为参考;直到收到第 4 帧的确认后,第 4 帧才被用作第 7 帧的参考。与 NAK 模式相比,ACK 模式的错误传播范围较小,但参考帧距离当前帧较远,因此编码效率较低。ACK 模式一般用于双程延时较短的情况。

MPEG 4 中也有一个与 RPS 类似的模式,称为 **NEWPRED**。在该模式中,帧间预测的参考帧随解码器返回的信息而自适应地调整。

9.5　差错控制小结

图 9-33 总结了我们所讲过的各种差错控制措施在多媒体通信系统中的位置,其中也包括网络低层的差错控制。数据链路层和物理层采用的差错控制措施包括 CRC、FEC、面向比特的数据交错和自动请求重传(ARQ)等,根据信道的不同各种网络采取的具体措施可能很不相同。图中的传输层打包策略应根据信道的错误率大小、有利于错误的检测和定位,以及解码的独立性等因素综合考虑。8.3.5节中对 H.223 的讨论则给出了在面向比特的网络上提高复用层容错性能的例子。图 9-33中的可伸缩编码(分层编码)和多描述编码将分别在 10.2 节和 10.4 节中讲述。分层编码可以看成是数据分割的一种方式,其中基本层码流比增强层码流对解码端图像的重建更为重要,重点保护基本层码流可以增强整个码流抵御错误的能力。图中所示的其他措施均在 9.3 节和 9.4 节中已经讲述。

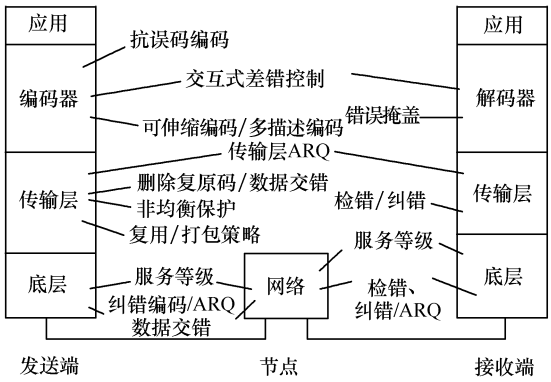


图 9-33　各种差错控制措施在系统中的位置

当存在接收端到发送端的反馈信道时，不仅可以进行如图所示的编解码器交互差错控制，发送端还可以根据反馈的当前网络状态信息（如错误率和可利用的带宽）进行**自适应**的差错控制，例如，在传输层自适应地调整包级别 FEC 的强度或 ARQ 的次数限制；在编码层自适应地调整编码参数（如帧内编码块数、插入再同步标记的间隔、帧间预测参考区域的限制等），以在当前网络条件下获得最优的性能。

在第 3 和 4 章中我们讨论了信源编码器的设计，即采用各种方法尽力压缩信源的码率，使已压缩码流中含有尽可能少的冗余信息；在 9.3 节中我们讨论了信道编码器的设计，即在压缩码流中加入冗余数据（FEC），以提高码流抵御传输差错的能力。经典的香农信息论告诉我们，只要信源以低于信道容量的速率表示，就可以独立地设计信源编码器和信道编码器，实现给定失真下压缩码流的无差错传输。这就是所谓的**分离定理**。但是分离定理存在的一个重要假设条件是信源编码和信道编码的长度是无穷大的，这从 3.3.3 节介绍的信源编码正定理中可以看出（N 充分大时（3-28）式成立），类似的条件在信道编码定理中也能看到。显然，在有实时要求的实际多媒体通信系统中，这个假设不能严格成立；换句话说，此时分开设计信源编码器和信道编码器不能真正达到分离定理所给出的性能。因此联合考虑信源和信道编码器的设计，即**信源—信道联合编码**吸引了不少学者的注意。目前在这方面的研究多数集中在保持信源和信道编码器各自独立但是联合优化它们的参数上，其中信源和信道编码之间的比特分配是讨论得较多的问题，即在信源编码速率与信道编码速率之和满足给定的信道速率的限制下，如何合理地分配信源编码和信道编码占用的比特数，以使端到端的总失真最小。在这里，总失真包括量化（信源编码）引起的和传输错误引起的重建图像失真两个部分。值得指出，目前信源—信道联合编码技术的研究主要针对的是比特错误，在包删除信道上针对压缩包流的研究成果还不是很多。

习　题　九

1. 为了与 TCP 友好，非 TCP 流在拥塞时按照 TCP 流量公式给出的速率来调整自己的流量大小，此时公式中的双程延时 RTT 和数据包大小 MSS 是下列情况的哪一种：(1)该流自己的 RTT 和 MSS；(2)该流所在网络上 TCP 流的 RTT 和 MSS。
2. 假设一个 128 kb/s 的压缩视频流在通过网络时，其 MSS＝1 500 B，RTT＝2s，使用 TFRC 进行拥塞控制（不考虑 min_rate 的作用）；
 (1)在传输过程中发现丢包率 $p=4\%$，速率应调整到多少才能保证对 TCP 的友好性？
 (2)在上述调整之后 5s，p 恢复到 0，此时速率应如何调整？作速率—时间图表示(1)和(2)的结果。
 (3)给出几种发送端实现码流速率改变的方法，并从复杂度和重建图像质量两个方面比较这些方法的优缺点。
3. 设网络上每个包的丢失率相同（均为 p）、且相互独立，采用 (n,k) 删除复原码，(1)推导解码后的残余丢包率(9-35)式。（提示：假设一个包丢失了，它能否被纠正呢？这取决于其他 $(n-1)$ 个包的情况。）

(2)若 N_p 代表连续 n 个包中丢失的包的个数，当 $N_p>(n-k)$ 时，丢失的包不可恢复，试根据 $P=E\{N_p\}/n$，其中 $E\{\}$ 代表均值，推导解码后的残余丢包率 P。

(3)当使用(7,4)码时，验证上述两式数值计算得到的结果相同。

4. 假设无线网络的误码率为 2×10^{-4}，进行 FEC 时使用的数据包长度为 100B(包括 RTP/UDP/IP 压缩包头和底层包头)，求：(1)网络丢包率(包中有一个错误即丢弃)；(2)使用(255,239)RS 码后的丢包率；(3)使用(32,16)缩短码后的丢包率；(4)这两种码的编码效率。

5. 用下列两种方式在 $g(x)=x^3+x+1$ 运算下的 GF(2^3)域中计算 $110\times101+010$，证明以下两种算法是等效的；

(1) 用生成元表示各数值，然后利用表 9.1 计算；

(2) 用多项式表示各数值，然后利用模 $g(x)$ 和与积计算。

6. 若 $g(x)=x^3+x^2+1$ 为 $GF(2)$ 上的 3 阶既约多项式，以 $g(x)$ 构成 $GF(2^3)$ 上的有限域，并找出该域中的本原元，将域中元素用 9.3.2 节中介绍的数字形式表示。

7. 设(7.4)二进汉明码的校验矩阵为

$$\boldsymbol{H}_1=\begin{bmatrix}0&0&0&1&1&1&1\\0&1&1&0&0&1&1\\1&0&1&0&1&0&1\end{bmatrix}$$

试导出系统码的生成矩阵，并证明所生成的汉明码与由(9-21)式所生成的码是等效的。

8. 已知一个码的域生成多项式为 $X^8+X^4+X^3+X^2+1$，码生成多项式为 $(X+\lambda)(X+\lambda^2)(X+\lambda^3)\cdots(X+\lambda^{15})$，请你

(1) 写出该码的生成矩阵；

(2) 该码的码长是多少？信息位是多少？可纠正多少个错误？

(3) 什么是缩短码？使用缩短码时，可纠正的错误个数是否会减少？

(4) 域生成多项式与码生成多项式有何联系与区别？根据域生成多项式求出在 GF(2^8) 上的域的元素有多少？

9. 对于由一个 $t=10$ 的(207,187)RS 码构成的序列进行 $i=52$ 的交错(见下图)，求该系统可纠正的突发错误的最大长度为多少 μs?



习题 9 图

10. 在抗误码编码中，阻止错误在时间域上传播的基本措施是下列方法中的哪一种(只选一个)：(1)使用定长编码；(2)插入同步头；(3)数据分割；(4)使用帧内编码；(5)采用非均衡保护。

11. 假设使用霍夫曼码传送下列字符串：ABACADABAC。(1)给出霍夫曼码树和相应的码表；(2)该码的编码效率是多少？为什么达不到 1？(3)如果该序列在传输中，第三个 A 的码字的第 1 位发生错误(0 变 1，或 1 变 0)，请写出解码后的字符串，能否检测出码字错误及错误的位置；(4)如果对字符串重新用可逆变长编码进行编码：{A:111;B:1011;C:10011;D:100011}，写出编码后的序列；(5)如果传输中第二个 A 的码字的第 1 位发生错误，标出发生错误的字段，并给出可正确解码的字符。

12. 设有(7.3)分组码，试问作为前向纠错码在噪声信道上使用时，能纠正错误的个数是多少？作为删除复原码在分组交换信道上使用时，能纠正的错误个数又是多少？解释二者相同或不同的原因。

13. 从编码效率和错误传播范围上对“参考图像选择”的两种方式(NAK 和 ACK)进行比较。在误码率较高的信道上传输时，你建议用哪种方式？

14. 说明 MPEG2 码流在误码率很低的环境下应用和在误码率较高的信道上传输的两种不同情况下，

(1) 每幅图像划分的 slice 应该多些还是少些？说明原因；

(2) 应采用 PES 复用还是 TS 复用？说明原因；

(3) 在误码率较高的信道上传输时，按相等数目的宏块来划分 slice 好，还是按相等数目的比特来划分 slice 好？为什么？

15. 一个帧率为 f 的实时 MPEG 流，为保证正常播放每帧数据应该以 $1/f$ 的间隔到达接收端。令 $0<r<1$ 为在规定时限内到达的帧的概率。假设发送端到接收端之间有一个 FIFO 路由器，路由器的带宽为 W、缓冲器大小为

B，在忽略链路延时和编解码延时的条件下，系统的延时 d 为 W、B 和 R 的函数，其中 R 为 MPEG 流的平均速率；网络的丢包率 c 也是 W、B 和 R 的函数。

(1) 分别给出 W、B 和 R 上升时，d 的变化趋势；

(2) 分别给出 W、B 和 R 上升时，c 的变化趋势；

(3) 显然，r 是 d 和 c 的函数，因此 r 也是 W、B 和 R 的函数，试给出 r 分别随 W、B 和 R 变化的趋势，并解释原因；

(4) 假设发送端进行 FEC，所加的冗余信息相当于平均每帧增加 h 比特，在帧率为 f 的情况下，送进网络的数据流的总速率 R' 是多少？

(5) 在 W 和 B 不变的条件下，FEC 的冗余加得越多是否越能使 r 上升？是否有条件限制？解释原因。

参 考 文 献

[1] J. Widmer, et al., "A Survey on TCP-friendly congestion control," IEEE Network, May/June., 2001, 28-37

[2] N. Hu and P. Steenkiste, "Evaluation and characterization of available bandwidth probing techniques," IEEE JSAC, Vol. 21, No. 6, 2003, 876-894

[3] M. Jain and C. Dovrolis, "End-to-end available bandwidth: measurement methodology, dynamics and relation with TCP throughput," IEEE/ACM Trans. on Networking, Vol. 11, No. 4, 2003, 537-549

[4] L. Rizzo, "On the feasibility of software FEC," http://www.iet.unipi.it/~luigi/softfec.ps

[5] J. Blomer, et al., "An XOR-based erasure-resilient coding scheme," Technical report, Inter. CS Institute, Berkeley, CA, 1995

[6] J. Li, "The efficient implementation of Reed-Solomon high rate erasure resilient codes," ICASSP, 2005, Ⅲ-1097-1100

[7] T. Paila, et al., FLUTE-File Delivery over Unidirectional Transport, RFC3926, 2004

[8] T. Whetten, et al., Reliable Multicast Transport Building Blocks for One-to-Many bulk-data Transfer, RFC 3048, 2001

[9] ISO/IEC 13818-2 Information technology-Generic coding of moving pictures and audio, Annex D, 1994

第 10 章　视频在异构环境中的传输

10.1　概述

在许多多媒体应用中，例如多媒体会议、IPTV 和点播电视等，发送端需要向具有不同 QoS 要求的多个用户提供同一种服务（如与同一个人交谈、或观看同一个节目），此时，如何有效地向这些用户提供不同质量的视频码流是本章讨论的主要问题。

为了对这个问题有进一步的了解，首先让我们考虑一个异构的网络环境，即在该网络中，某部分子网的速率高，而另一部分子网的速率低。对于传统的数据传输（例如文件下载）而言，子网的速率只影响数据传输的时间，不管通过低速网还是通过高速网下载连接在主干网上的某个服务器内的文件，所得到的文件是相同的，只是下载的时间长短不同而已。而当分别连接在高速和低速子网上的两个终端之间进行点对点的多媒体通信时，在通信正式开始之前二者可以进行 QoS 协商，然后在一个商定的较低的 QoS 水平（即低速率）上相互通信。在上述两种情况中，发送端都只需要具有一种质量（码率）的已压缩视频。但是在**一点对多点**的多媒体通信中，例如服务器同时向两个接入速率不同的终端传送同一节目，或者多媒体会议中连接在高速子网上的发送端，既需要实时地向连接在高速子网上的与会者传送高速码流，又需要同时向连接在低速子网上的与会者传送低质量（低速率）码流。在这种情况下，如果发送端只提供一种质量（一种码率）的视频流就不能满足不同用户的需要了。例如服务器只提供 800kb/s 的视频码流，通过局域网接入的用户甲可以获得良好的视频观看质量，而通过 512kb/s 的 ADSL 接入的用户乙则由于带宽的限制而不能获得连续和稳定的视频图像。这时如何将一个发送码流“剪裁”到满足不同接收端所要求的质量就是必须考虑的了。通常，完成 QoS“剪裁”功能的实体称为**QoS 过滤器**（Filter）。过滤（即已编码流的速率转换）一般在高速和低速子网的衔接处（网关）进行。在异构网络中，过滤功能可以进行多次。图 10-1 给出了使用 2 次过滤的例子。

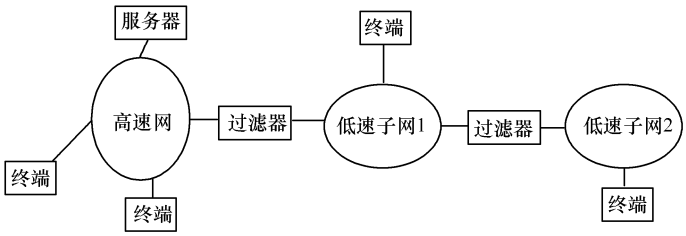


图 10-1　已编码流的过滤

除了网络条件不同以外，造成接收端 QoS 要求不同的原因还有终端处理能力和显示能力的不同（例如手机的存储和 CPU 处理能力低，只能显示小尺寸的图像，而 PC 的存储和处理能力高、能显示高清晰度的图像等），以及应用的需求不同和愿意承担的通信费用不同等。不同性能（带宽、延时等）或使用不同协议的网络和具有不同 QoS 要求的终端结合在一起则构成我们所说的异构环境。

除了异构环境外，在不能提供稳定 QoS 的网络（如 IP 网）上，可能需要动态地调整视频的速率，以适应当前的网络带宽（如在 8.8 节的动态自适应 HTTP 流系统中）；或者当网络负荷过重时，需要降低发送的视频流速率，以避免过多的丢包（如在 9.2 节的拥塞控制中），也需要进行码流速率的

转换。

最简单的提供多种质量的视频流的方法是在发送端用多个视频编码器对同一视频序列进行编码，得到多个不同质量的视频流，例如形成 8.6.4 节中的多速率流媒体文件。但是对同一视频内容编码形成的多个独立的视频流之间存在大量的冗余信息，因此在存储和传输时将占用相当多的存储空间和带宽资源。另外，随着新业务的不断涌现，预编的视频流也难以保证能够适应新的视频业务的需求。

获得不同速率码流的另一个简单办法是，通过所谓的**丢帧过滤器**丢弃掉码流中的 B 帧 /P 帧，将输出码流所需的带宽降低。但这种方法只能改变帧率，且不一定能保证相等的帧间隔，因此在速率变化较大时，重建图像质量不高。

第 3 种提供多种质量视频流的方法是对视频序列进行**可伸缩性视频编码 SVC**(Scalable Video Coding)。所谓可伸缩性编码是指，对一个高质量的视频序列进行一次编码，但允许从该已编码流中截取出某一部分进行解码得到质量比原视频低的序列，而对整个码流解码则得到与原视频质量基本相同的序列来。可伸缩性视频编码为视频流在异构环境中的传输提供了灵活的方式，此时网关处的 QoS 过滤、或低能力终端的接收都变得十分简单，只需要从码流中截取适当的部分就可以了；但是 SVC 与一般的(非可伸缩性) 编码方式相比编码效率要低，例如对于相同的视频质量，SVC 码流的速率比一般编码码流可能高 10% ～ 50%，甚至于更多[1]。

第 4 种提供多种质量的视频流的方法是采用**视频转码**(Transcoding) 技术，即在 QoS 要求发生改变的节点处(如图 10-1 的过滤器) 将已压缩视频流转换成适合于新 QoS 要求的视频流。视频转码包括已压缩视频流在不同码率、不同空间分辨率、不同帧率和不同压缩标准之间的转换。视频转码的算法复杂度比可伸缩性视频编码要低，是目前工程上较易实现的一项技术。

我们将在 10.2 和 10.3 节中分别对可伸缩性视频编码和视频转码进行讨论。在 10.4 节中我们将介绍另一种编码方式，称为**多描述编码 MDC**(Multiple Description Coding)。多描述编码也可以在异构环境中应用，但它更重要的应用还是在噪声信道上的传输。

10.2 可伸缩性编码

10.2.1 可伸缩性编码的概念

可伸缩性编码可以通过**分层编码**(Layered Coding)，也可以通过**嵌入式编码**来实现。图 10-2 给出了分层编码、解码系统的方框图。图中基本层码流给出低质量的图像；增强层可以有多层(见图中虚线所示)，基本层和增强层码流的逐层结合给出质量逐级递增的图像，即分层编码器的输出支持数种质量的服务，而解码器通过选择码流层次不同的子集可以将图像质量“剪裁”到所需要的水平。在这里，图像质量不只限于从空间分辨率(大尺寸图像还是小尺寸图像) 的角度评价，也可以从帧率或量化误差来评价。因此可伸缩性编码包括**空域**(分辨率) 可伸缩性、**时域**(帧率) 可伸缩性和**质量**(信噪比) 可伸缩性 3 个维度。

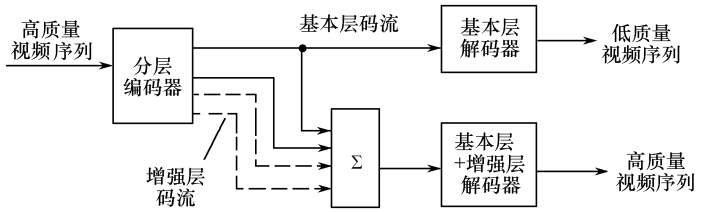


图 10-2 分层编码系统

可伸缩性编码不仅如前节所述，可以在异构环境中提供视频传输的灵活性，还可以用来在传输条件变坏时保持系统性能在可接受的水平。例如当网络发生拥塞时，网络节点可能要被迫丢弃一些数据。在一般编码的码流中，数据的丢失不仅造成丢失部位图像的缺损，而且误差还会在空间上和时间上扩散（见 9.4.1 节）。如果采用分层编码，节点在拥塞时只丢弃增强层的数据，那么接收端通过接收基本层数据仍能收到完整的图像，只不过质量降低一些而已。同时，分层编码可以看成是一种数据分割的方式，基本层数据比增强层有更高的重要性；对基本层和增强层进行不均衡保护，能够在强噪声条件下保持完整图像的接收，尽管此时的质量有所降低。

另一个可能应用分层编码的例子是在数字电视的地面或卫星广播中。通常对已压缩的码流采用前向纠错码（FEC）来提高其抗误码的能力，但是 FEC 所能纠正的错误个数是有限的，当接收条件恶化（例如在广播复盖区的边沿地带）时，传输错误只要比 FEC 所能纠正的错误多一个，FEC 就不再起作用，解码的图像质量将突然下降，可能到达不可接受的程度，这通常称为**数字截断**（Digital Cutoff）。而在模拟电视广播中，则不会发生这种情况，随着信噪比的下降（接收机靠近复盖区的边沿）图像质量是逐渐下降的。图 10-3 表示出在这两种情况下图像质量下降的情况。如果采用分层编码，将基本层数据用低误码信道（如大功率发射，或使用更强的 FEC）传送，而增强层数据用高误码信道（如低功率发射、或简单的 FEC）传送，那么随着信噪比的降低，增强层数据虽然不能很好地被接收，但基本层数据仍能被接收，并给出虽然质量不太好、但是完整的图像，即可以近似像模拟电视广播那样实现接收质量的逐步下降。

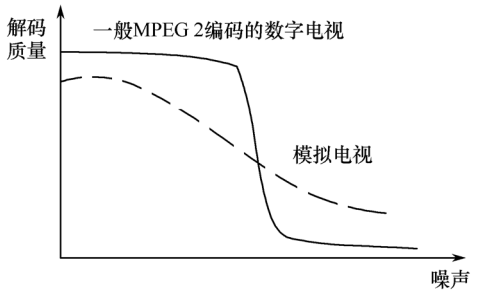


图 10-3　噪声对解码图像质量的影响

10.2.2　空间域可伸缩性编码

采用空间域可伸缩性编码的目的在于提供空间分辨率不同的分层编码码流。在这种编码方式中，原始图像在水平和垂直方向上分别进行下取样得到低分辨率（一般是原图像的 1/4）的图像，然后对该图像再进行下取样，将分辨率进一步降低，如此重复，直到得到所需要的最低分辨率图像。如图 10-4 所示，不同尺寸的图像构成一个金字塔，最低分辨率的图像（金字塔顶层）作为基本层进行编码，其余不同分辨率的图像分别作为增强层编码。

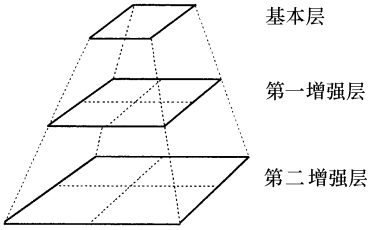


图 10-4　空间分辨率不同的图像

MPEG 2、H.263 和 MPEG 4 都支持两层（基本层和一个增强层）的空间可伸缩性编码，其原理方框图如图 10-5 所示。原始图像经下取样后得到基本层图像，对基本层图像按一般编码方式进行编码。另一方面，解码后的当前基本层图像经内插构成当前高分辨率图像的预测值。与此同时，像在一般编码中一样，前一个已编码的高分辨率图像经运动补偿后也产生一个当前高分辨率帧的预测值。在每个宏块进行编码时，选择两个预测值中产生预测误差较小的一个、或两个预测值的平均值作为该宏块进行预测编码的参考。解码器采用与编码器相同的参考宏块，因此由参考宏块和从增强层码流解码得到的预测误差之和，可以正确地恢复出高分辨率图像来。

在 H.264 可伸缩性扩展中（该扩展于 2004 年被 MPEG 组织选择作为 MPEG SVC 项目的工作起点），为了进一步利用层间的相关信息以提高编码效率，分辨率可伸缩性编码增加了 3 种层间预测模式：

（1）帧内编码宏块纹理的层间预测

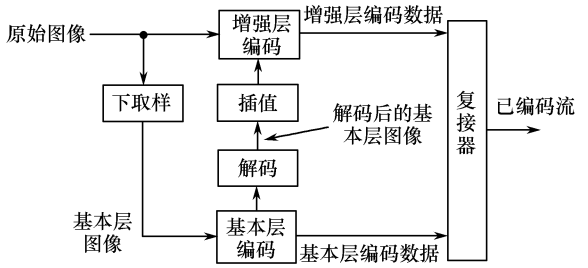


图 10-5　空间域可伸缩性编码原理方框图

当增强层宏块使用帧内预测模式进行编码时，除采用本层帧内预测模式外，还定义了 Intra_Base 宏块模式。Intra_Base 宏块采用低分辨率层相对应位置宏块对增强层宏块进行纹理预测。在预测之前，首先需要对低分辨率图像宏块进行去块效应滤波，以消除宏块内部边界和外部边界的块效应；然后再对宏块各方向进行 4 像素的边界延拓；最后，根据低层与高层分辨率之比进行插值得到预测纹理信息。

(2) 运动信息的层间预测

在对增强层运动信息编码时，除 H.264/AVC 规定的运动预测模式外，还增加了两种层间运动信息预测模式：Base_Layer_Mode 和 Quarter_Pel_Refinement_Mode。

当宏块采用 Base_Layer_Mode 预测模式时，宏块的运动信息(运动矢量、宏块分割模式以及参考帧序号) 完全采用低层对应宏块的运动信息，不再进行额外的运动信息编码。由于当前层与低层存在下采样过程，其宏块的运动矢量及分割模式也应根据比例进行缩放。例如：当分辨率的伸缩为常见的 1/2 下采样时，低层的运动矢量均扩大为原来的 2 倍。而当低层对应宏块分割模式采用 direct、16×16、16×8、8×16 和 8×8 时，当前层宏块均采用 16×16 分割模式；当低层对应宏块分割模式为 8×4、4×8 或 4×4 时，当前层宏块分别采用 16×8、8×16 和 8×8 分割模式；当低层对应宏块采用帧内编码模式时，当前层宏块采用 Intra_Base 模式进行帧内编码宏块纹理的层间预测。

当宏块采用 Quarter_Pel_Refinement_Mode 预测模式时，宏块分割模式及参考帧序号的预测与 Base_Layer_Mode 模式相同，其差别主要在运动矢量的预测上。Quarter_Pel_ Refinement_ Mode 模式对从低层获得的运动矢量在 x 和 y 方向上各编码一个 1/4 像素精度的修正量{−1,0,1}，将修正量加上低层获得的运动矢量后得到当前层宏块的运动矢量。此模式仅在低层分辨率低于当前层分辨率时有效。

(3) 帧间编码宏块残差信息的层间预测

对于帧间编码的宏块，由于各层运动信息之间具有较强的相关性，因此宏块经运动补偿之后得到的残差信息也具有较强的相关性，有时候对残差信息做层间预测也能提高编码效率。但是对残差信息的层间预测并不总是能提高编码效率的，如果当前层运动信息与低层运动信息差别较大时，由于不同层的预测宏块差别较大，导致各层间残差信息相关性降低，这时对残差信息采用层间预测将降低编码效率。只有当各层运动信息一致或接近一致时，对残差信息采用层间预测才能提高编码效率。当前宏块的残差信息是否采用层间预测模式用 residual_prediction_ flag 标志表示。

如果当前层与低层分辨率不同时，需要对低层残差信息进行像素插值。如果分辨率之比为 1∶2，采用插值滤波器[1,1]/2 进行插值，其他情况下则采用 H.264/AVC 规定的 1/4 像素插值方式。

10.2.3　时间域可伸缩性编码

采用时间域可伸缩性编码的目的是提供帧率在一定范围内变化的分层编码的码流。MPEG 2、H.263 和 MPEG 4 支持 2 层时域可伸缩性编码。其具体做法是，首先将原始图像序列沿时间轴进行

隔帧(2∶1)的抽取,抽取后得到的低帧率序列按一般方式编码得到基本层码流,剩余的帧再编码构成增强层码流。增强层编码器与一般的编码器也类似,只是在进行运动补偿时可以采用增强层图像、或采用重建的基本层图像作为参考帧。

H.264 可伸缩性扩展提供如下两种可选模式来实现时域可伸缩性编码。

1. 分级 B 帧结构

由于预测编码采用闭环结构,视频流中丢弃I帧和P帧都将引起后续帧的解码误差并导致误差扩散。为了避免这种情况的发生,解码时所能丢弃的帧只有 B 帧。传统的 B 帧不作为其他帧的参考图像,因此如果采用传统的预测编码结构,帧率的改变只有一级可改,例如 IBPBP… 的 30 帧 / 秒序列结构,在帧率上只能降为 IPP… 的 15 帧 / 秒的序列。为了增加帧率可伸缩的层次,在 H.264 可伸缩性扩展中提出了分级 B 帧的概念,第 i 级 B 帧 B_i 可以作为第 $i+1$ 级 B 帧 B_{i+1} 的参考帧。图 10-6 即为可实现 4 级帧率可伸缩的可分级 B 帧预测模型。调整帧率时,通过依次丢弃 GOP 中的 B_3、B_2、B_1 帧,便可将帧率成 2 倍地下降。

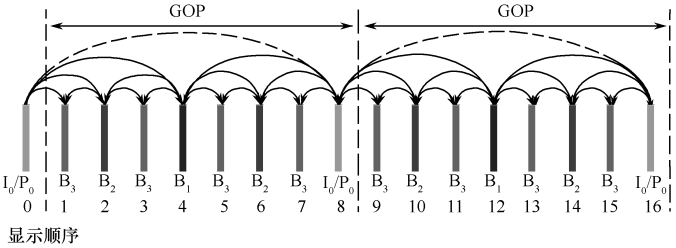


图 10-6　分级 B 帧的 GOP 结构

2. 基于运动补偿的时域滤波(MCTF)分解

在信号分解中,**提升小波**在保持完全重构信号方面具有结构上的优势,无论进行线性或非线性滤波,提升小波都能容易地实现信号的完全重构,因此 **MCTF**(Motion Compensated Temporal Filtering)分解采用基于提升小波的开环编码结构。在可伸缩性编码中,开环编码结构与闭环编码结构相比,它采用原始图像作为预测图像,能够有效的避免预测误差扩散。

MCTF 分解主要分三个步骤。

① 分解:序列图像 $S[k]$ 在时间上分解为奇帧 $S[2k+1]$ 和偶帧 $S[2k]$。

② 预测:预测过程是将偶帧图像 $S[2k+2r_i]$ 的线性组合作为奇帧图像 $S[2k+1]$ 的预测图像 $P(S[2k])$,将 $S[2k+1]$ 与 $P(S[2k])$ 相减得到高频信息图像 $H[k]=S[2k+1]-P(S[2k])$。

由于序列图像在时间域上的强相关性是基于运动的,因此预测图像 $P(S[2k])$ 也是基于运动的,即

$$P(S[2k])=\sum_{i=1}^{t}\alpha_i S[x-MV_x,y-MV_y,2k+2r_i] \tag{10-1}$$

其中 $\sum_{i=1}^{t}\alpha_i=1$,(MV_x,MV_y) 为像素点 (x,y) 的运动矢量。

在式(10-1)中,当 $t=1$,$r_1=0$ 时,MCTF 采用 Haar 滤波器,其预测图像 $P_{\text{Haar}}(S[2k])$ 为

$$P_{\text{Haar}}(S[2k])=S[x-MV_x,y-MV_y,2k] \tag{10-2}$$

对于 Haar 滤波器,预测过程相当于 H.264 中的 P 帧预测模式。

在式(10-1)中,当 $t=2$,$\alpha_1=\alpha_2=1/2$,$r_1=0$,$r_2=1$ 时,MCTF 采用 5/3 滤波器组,其预测图像 $P_{5/3}(S[2k])$ 为

$$P_{5/3}(S[2k])=\frac{1}{2}(S[x-MV_x,y-MV_y,2k]+S[x-MV_x,y-MV_y,2k+2] \tag{10-3}$$

对于 5/3 滤波器，其预测过程相当于 H.264 中的 B 帧预测模式。

③ 更新：更新过程是将高通信息图像 $H[k-r_i]$ 的线性组合作为更新图像 $U(H[k])$ 加入到偶帧图像 $S[2k]$ 中，得到低频信号图像 $L[k]=S[2k]+U(H[k])$。与预测过程相似，其更新图像也是基于运动的，即

$$U(H[k])=\sum_{i=1}^{s}\beta_i H[x-\tilde{MV}_x,y-\tilde{MV}_y,k+\tilde{r}_i] \tag{10-4}$$

其中 $(\tilde{MV}_x,\tilde{MV}_y)$ 为 (x,y) 的运动矢量，可通过预测过程得到的运动矢量反解得到。

与预测过程相对应，若 MCTF 采用 Haar 滤波器，则取 $s=1,\beta_1=1/2,\tilde{r}_1=0$，其更新图像 $U_{\text{Haar}}(H[k])$ 为

$$U_{\text{Haar}}(H[k])=\frac{1}{2}H[x-\tilde{MV}_x,y-\tilde{MV}_y,k] \tag{10-5}$$

若 MCTF 采用 5/3 滤波器，则取 $s=2,\beta_1=\beta_2=1/4,r_1=-1,r_2=0$，其更新图像 $U_{5/3}(H[k])$ 为

$$U_{5/3}(H[k])=\frac{1}{4}(H[x-\tilde{MV}_x,y-\tilde{MV}_y,k-1]+H[x-\tilde{MV}_x,y-\tilde{MV}_y,k]) \tag{10-6}$$

序列图像经过一次 MCTF 分解后，得到一组低通图像 $L[k]$ 和一组高通图像 $H[k]$，在视频解码时，若只解码低通图像，可以得到帧率降为一半的视频序列。一般的，对视频序列做 3 次 MCTF 分解可实现 4 级帧率可伸缩性码流。

10.2.4 质量可伸缩性编码

视频编码的客观质量一般用解码重构图像的峰值信噪比来衡量，因此质量可伸缩性也称为**信噪比可伸缩性**，它提供不同视觉质量的分层编码的码流。

MPEG 2 和 H.263 支持 2 层的信噪比可伸缩编码，图 10-7 给出了这种模式的简要方框图。在基本层中，DCT 系数用一个粗量化器（量化步长大）量化，量化后的系数构成基本层数据；然后对粗量化后的 DCT 系数和量化之前的实际值之差（即量化误差）再进一步用一个细量化器进行量化，则构成增强层的数据。普通的 MPEG 2 解码器可以对基本层码流解码得到信噪比较低的图像。要得到高信噪比的图像必须在解码过程中将增强层携带的 DCT 系数的误差值加到基本层的粗量化 DCT 系数上，再进行 IDCT。一般地，利用这种方式产生少量的层数，因而码率只有几种变化，通常称为**粗粒度** SNR 可伸缩性。

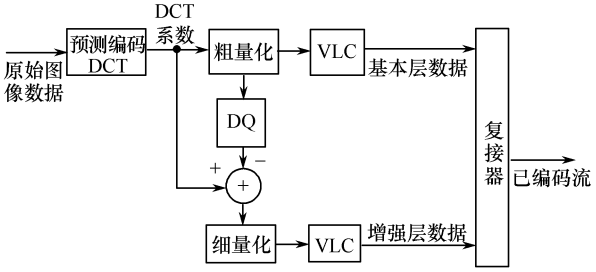


图 10-7　信噪比可伸缩码器简单方框图

MPEG 4 提出了一种实现**细粒度** SNR 可伸缩性（Fine Granularity Scalability，**FGS**）的方法。在这种方法中，增强层的数据（DCT 系数的残差）采用一个很小的量化步长量化，然后进行**位平面编码**，即将每个宏块量化后的系数残差值以二进制表示，首先从最重要比特 MSB（Most Significant Bit）平面进行变长编码，然后依次进行其余平面的编码，直到最不重要比特 LSB（Least Significant Bit）平面。这样形成的增强层码流可以在多个点（对应不同的比特平面）截断，从而提供质量（码

率）变化的更大灵活性。

H.264 可伸缩性扩展既支持粗粒度、也支持细粒度的 SNR 可伸缩性编码。在 H.264 可伸缩性扩展的 FGS 中，增强层对原始图像与前一质量重建图像之间的差值图像进行编码，每一个增强层数据的量化步长为前一层量化步长的一半，这称为**渐次增进**(Progressive Refine)。增强层码流可以在任意点被截断，以得到逐步精细的视频图像。为了实现这一目标，需要将数据按重要性程度排序，重要数据优先被扫描编码。因此不同于H.264/AVC 基于变换块的 Z 字形扫描方式，FGS 采用基于全帧的扫描方式，按变换系数的频带顺序扫描系数，即 FGS 首先扫描全帧所有块的亮度 DC 系数及扫描色度 DC 系数，然后再扫描所有块的第 1 个 AC 系数，如此下去直至结束。同时将变换系数分为重要系数和非重要系数，进行两次扫描：

第一遍扫描编码（重要性扫描）：对所有在基本层和前一增强层中对应位置系数为 0 的系数（称为非重要性变换系数）进行扫描编码；

第二遍扫描编码（精细化扫描）：对所有在基本层和前一增强层中对应位置系数非 0 的系数（称为重要性变换系数）进行扫描编码。

10.2.5 可伸缩性编码的联合应用

1. MPEG 2 可伸缩性编码联合应用

时域、空域和质量可伸缩性编码可以联合使用构成更多层次的分层编码。例如图 10-8(a) 和 (b) 给出 MPEG 2 中的 2 个例子。在(a) 中，首先将输入的高清晰度电视（HDTV）信号经下取样成普通电视信号，再用信噪比可伸缩编码器将普通电视信号编码形成基本层（0 层）和第 1 个增强层（1 层）的数据。然后对 0 层和 1 层数据共同解码得到的高信噪比的普通电视信号进行内插，作为输入的高清晰度电视信号的预测值，预测误差值编码形成的即为第 2 增强层（2 层）的码流。

层次	可伸缩性	应　用
0		普通电视
1	信噪比	高质量电视
2	空间域	HDTV

(a)

层次	可伸缩性	应　用
0		普通电视
1	空间域	隔行扫描的 HDTV
2	时间域	逐行扫描的 HDTV

(b)

图 10-8　MPEG 2 中可伸缩性编码的联合应用

在(b) 中，输入信号为逐行扫描的 HDTV 信号。从该信号的每 2 帧中抽取 1 帧组成的序列其帧率降低了一半，相当于一个隔行扫描的 HDTV 信号。对这个低帧率序列进行空间域的可伸缩性编码，其基本层（0 层）提供普通电视（低分辨率、低帧率）质量的图像，而 0 层和 1 层数据提供低帧率的 HDTV 图像。如果将第 2 层数据解码，并将得到的序列与由 0 和 1 层数据解码得到的序列复接在一起，就能恢复出高帧率（逐行扫描）的 HDTV 序列来。

2. JSVM 编码器结构

针对 H.264 可伸缩性扩展，由 MPEG 和国际电联组成的联合视频小组（Joint Video Team，JVT）提出了一个**联合可伸缩视频模型 JSVM**(Joint Scalable Video Model)[1]。JSVM 是一个基于 DCT 变换的可伸缩视频编解码框架（见图 10-9），它同时可以实现 3 个维度的视频可伸缩性，提供质量、帧率和分辨率上可伸缩的码流。编码器先对视频序列进行空间下采样，得到不同层次分辨率的图像序列，然后对每一层次分辨率序列图像依次（从低分辨率到较高分辨率）进行基于 H.264/AVC 的核心编码。在对每一分辨率图像序列编码时，先对图像序列进行时间域上的图像分解，以实现时域上的可伸缩性。同时对同一帧率和分辨率的视频序列采用粗粒度、或渐次增进(FGS)SNR 可伸缩性编码。

由于存在多个层次的分辨率序列，此时的 H.264/AVC 预测编码将支持层间预测技术。

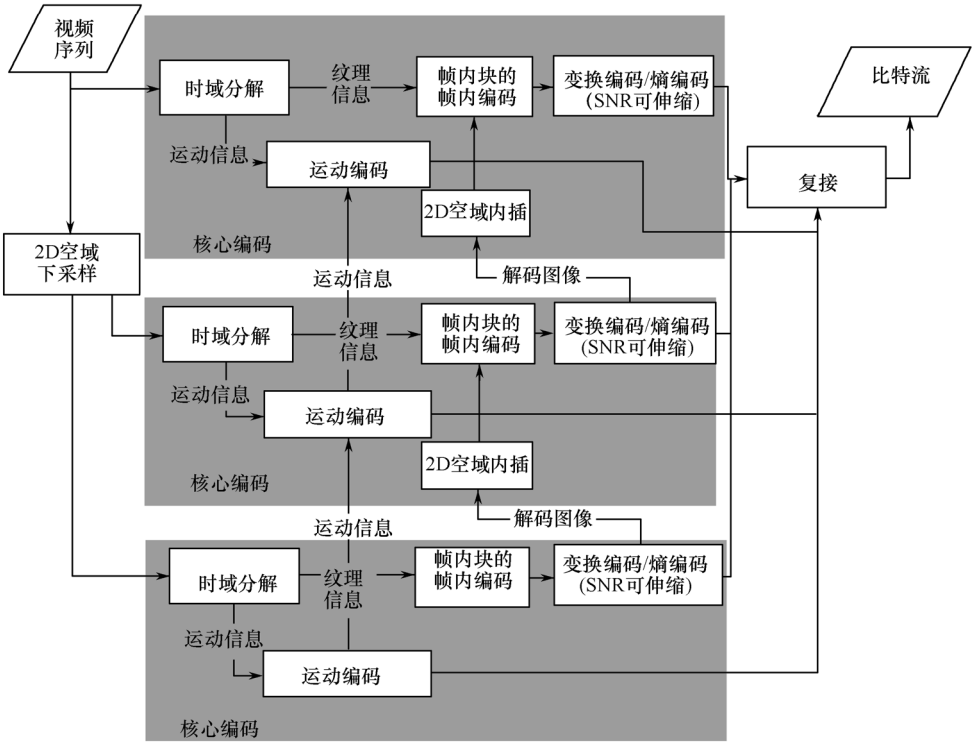


图 10-9　JSVM 编码器结构

10.2.6　基于小波变换的可伸缩性编码

小波变换是一种时频局部化变换，它将信号分解到不同的尺度空间上进行处理，这种思想完全与可伸缩性编码吻合。基于小波变换的可伸缩性视频编码框架具有更灵活的可伸缩性，目前也越来越受到重视[2]。3.8 节介绍的 EZW 算法、SPIHT 算法和 EBCOT 算法都是适合静态图像的一种可伸缩性编码（嵌入式编码）。目前基于小波变换的视频编码框架，也都符合可伸缩性视频编码的要求。对于视频序列，小波变换增加了时间维度，由于序列图像在时间上的相关性是基于运动的，因此在时间维度上，也往往采用前述的 MCTF 技术实现。

10.3　视频转码

视频转码（Video Transcoding）是指将某种编码格式的视频压缩码流转换成另一种编码格式的视频压缩码流。视频转码主要实现下述 4 种视频流的转换：

（1）码率转换：通过对变换系数二次量化等速率控制技术可以将高码率的视频流转换成适合当前信道带宽的低码率的视频流；

（2）空间分辨率转换：通过下采样与上采样技术可以将当前视频序列的分辨率进行缩小或放大；

（3）时间分辨率转换：通过抽帧减少当前视频序列的帧率；

（4）语法转换：将符合某种压缩标准句法的视频流转换成另一种压缩标准的视频流。

上述 4 种功能也可以进行灵活的组合以符合实际应用的需求。

与可伸缩性视频编码相比，视频转码技术将视频流转换成适合异构网络传输及采用不同解码标

准终端的视频流，终端用户不需要更换新的解码器就可以获得合适的视频流。而可伸缩性视频编码需要终端用户使用可伸缩性视频解码器，这将增加部分终端用户的额外开销。尤其对于某些低功耗的便携式终端，可伸缩性视频解码器也会增加电能消耗，这将导致设备持续使用时间的降低。因此从工程应用的角度来看，视频转码技术比可伸缩性视频编码技术更具备产品化优势。

10.3.1 视频转码框架

视频转码是将某种编码格式视频流转换成另一种编码格式视频流的一个工具，只要输入流和输出流符合一定的标准，视频转码器就能与其他视频应用工具完全兼容，因此，视频转码存在多种可选的框架，但其实质就是一个视频解码器与一个视频编码器在不同程度上的级联。

1. 开环视频转码框架

开环视频转码对输入视频流先进行变长解码和反量化，再根据新目标码率调整量化台阶，重新进行量化和变长编码，得到输出视频流。开环视频转码框架如图 10-10 所示。这种转码框架省去了解、编码的 DCT/ 逆 DCT 变换、运动补偿等模块，所以运算复杂度非常低。另外，转码时解出一个宏块即可进行再量化和编码，所以也不需要任何帧存，转码延迟小、所需要的存储资源也非常小。

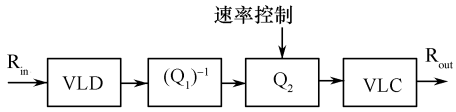


图 10-10　开环转码框架

但是这种开环结构中解码参考帧和编码参考帧不一致，从而引起误差漂移，这种误差将不断累积传递，直到出现帧内编码的帧或宏块。如果相邻两个帧内编码帧之间的间隔很大的话，将导致图像质量严重下降。而且如果在开环转码结构中进行帧率和分辨率转换的话，这种误差的累积传递将更大，因此开环转码结构一般适用于转码设备简单、仅做码率转换且对图像质量要求不高的情形。

2. 闭环视频转码框架

闭环视频转码主要解决开环转码框架中由于解码器参考帧和编码器参考帧不一致而引起的误差漂移问题，它在处理残差帧信息时增加了运动补偿模块，以避免由于两端参考帧不一致而引起的误差漂移。根据视频转码中运动补偿模块接口数据的不同，闭环视频转码框架有多种运算复杂度不同的框架。

(1) 完全的像素域解编码器级联框架

完全的像素域解、编码器级联框架将输入的视频压缩流进行完全解码，再重新编码生成新的符合要求(指定码率、帧率、分辨率和标准格式)的视频压缩流，如图 10-11 所示。在这个框架中，解码端将输入视频流进行完全解码得到原始图像，然后进行相应的编码预处理(如分辨率转换、帧率转换等)，最后进入编码器进行独立的编码，因此该框架具有高度的灵活性，可以实现视频转码的所有需求。同时，由于解码和编码完全独立，利用该转码框架进行转码不会引起误差漂移，转码后所得的视频主、客观质量一般也是最优的。实现完全的像素域解编码器级联框架也相对比较容易，只需将解码器、编码预处理和编码器简单连接即可，不需要增加新的模块。

然而完全的像素域解编码器级联框架也是运算复杂度很高的转码框架，因为它包括了完整的解码和编码所需的运算，而解码器中的 IDCT 变换、编码器中的 DCT 和 IDCT 变换、运动估计是计算量很大的模块，尤其是运动估计模块，约占整个编码器运算量的 60%。另外，由于解码和编码完全独立，视频数据需要在解码器端和编码器端分别缓存，对存储资源的要求也比较高。

与视频编码不同，视频转码的输入视频流已经是经过视频压缩之后的码流，因此在输入数据中本身带有视频压缩的各种有用信息，如运动信息、DCT 变换后的系数以及速率控制参数等。这些信息在视频转码的编码端没有必要再去计算一遍，可以合理地利用输入端信息对完全的像素域解编

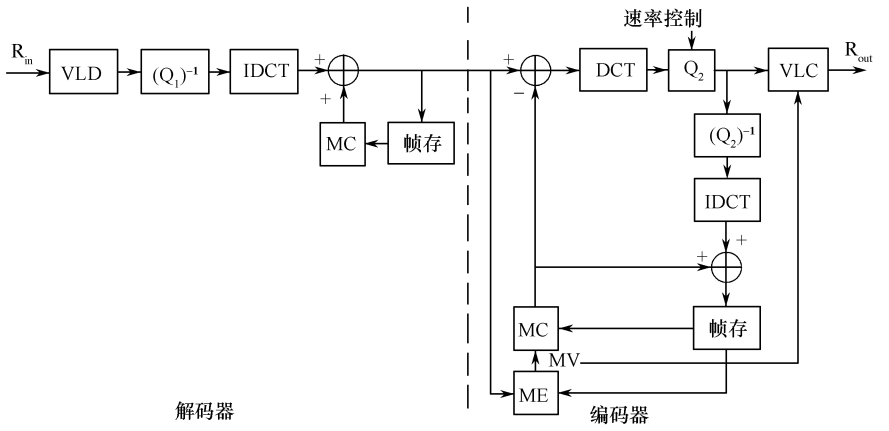


图 10-11　像素域级联转码框架

码器级联框架进行适当的简化。

(2) 基于像素域运动补偿的视频转码框架

基于像素域运动补偿的视频转码框架对输入流进行可变长解码、反量化、逆 DCT 变换和反运动补偿，得到完全重构的图像数据，然后根据转码要求进行编码预处理(分辨率转换、帧率转换等)，最后利用原有的编码信息(帧类型、运动信息等)和速率控制进行运动补偿、DCT 变换、量化和可变长编码得到输出视频流，如图 10-12 所示。与完全的像素域解编码级联框架相比，该转码框架减少了编码端运算复杂度非常高的运动估计模块，提高了视频转码的速度。

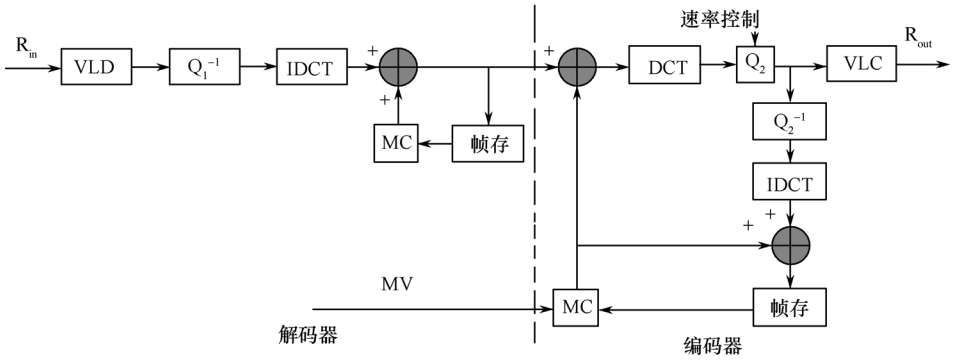


图 10-12　基于像素域运动补偿的视频转码框架

然而，当视频转码中宏块类型、帧率或分辨率发生变化时，序列图像在时间上的相关性将发生变化，因此输入端的运动矢量可能并不是最优的，此时完全使用原有的运动信息将会造成图像质量的下降。为了弥补质量损失，运动矢量再利用模块需要进行简单的运动矢量再估计以修正原有运动矢量，减少图像质量损失。

(3) 基于 DCT 域运动补偿的视频转码框架

基于 DCT 域运动补偿的视频转码框架对输入流进行可变长解码、反量化和 DCT 域反运动补偿，得到 DCT 域的图像数据，然后根据转码要求进行编码预处理(分辨率转换、帧率转换等)，最后利用原有的编码信息和速率控制进行 DCT 域运动补偿、量化和可变长编码得到输出视频流，如图 10-13 所示。与基于像素域运动补偿的视频转码框架相比，该转码框架采用 DCT 域运动补偿技术来替代 DCT/ 反 DCT 变换和像素域运动补偿模块，该技术利用残差信息的 DCT 系数的稀疏特性，可以有效的降低运算复杂度，从而提高转码速度。

当然采用 DCT 域运动补偿的视频转码框架也存在一定的局限性，在某些转码需求下（如支持去块效应技术或分辨率转换），精确的 DCT 域上的运动补偿运算量也非常高，因此，需要采用一些逼近算法来近似计算 DCT 域运动补偿，目前这方面的研究还不多。

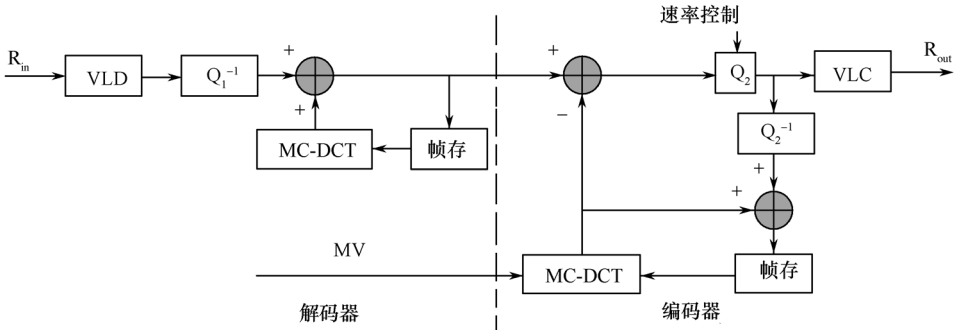


图 10-13　基于 DCT 域运动补偿的视频转码框架

10.3.2　运动向量的再利用

视频转码应该充分的利用输入视频流的编码信息（如运动信息、纹理信息和速率控制信息等），在编码端相应的降低运动估计、纹理信息编码和速率控制等的运算复杂度，从而达到在保持视频图像主、客观质量的要求下提高转码速度的目的。因此视频转码的关键技术也主要集中在对原有编码信息的利用上，主要包括：运动向量的再利用、DCT 域运动补偿（MC-DCT）、速率控制的再量化等。

基于运动补偿的帧间预测是去除序列图像时间冗余度的一项关键技术，一般的，运动估计越精确，预测后的残差信息能量越少，视频编码效率也就越高。然而运动估计也是视频编码中最耗计算资源的，其计算量往往占整个编码的 60% 以上。在视频转码中，通过充分利用输入视频流的运动信息，我们可以尽可能地避免复杂的运动估计，以减少资源开销和提高转码速度。在简单的转码情况下（如相同压缩标准的视频流仅做码率转换），其运动信息可以在转码中完全重用。然而对于一些较为复杂的转码，原输入码流的运动信息往往需要进行一定的重估计和校正，这些技术也就是运动向量再利用技术。

1. 空间分辨率改变下的运动矢量映射

在基于块匹配的运动估计中，块的尺寸一般为 16 × 16 的宏块。当视频图像空间分辨率发生改变时，其宏块的划分也随之发生改变，因此输出视频流的运动信息将做相应的变换。例如在常见的视频图像 2∶1 下采样过程中，原视频图像相邻的 4 个宏块经下采样后变成一个宏块，此时需要从原来 4 个运动矢量中获得一个运动矢量（如图 10-14 所示）。常见的方法主要有：

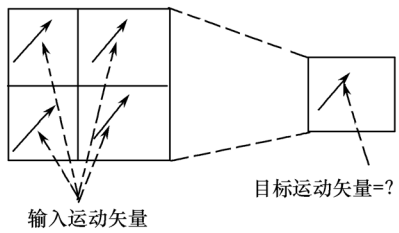


图 10-14　2∶1 下采样下的 MV 映射

（1）自适应运动矢量下采样方法[3]

设输入运动矢量分别为 mv_i，$i=1,2,3,4$，输出运动矢量为 mv，则可将这 4 个输入运动矢量进行加权平均得到 mv，计算公式如下：

$$mv = \frac{1}{2}\sum_{i=1}^{4} mv_i A_i \Big/ \sum_{i=1}^{4} A_i \tag{10-7}$$

其中 A_i 为 P 帧第 i 个宏块残差数据中非零 AC 系数的个数。

（2）主运动矢量方法

按一定的准则在4个运动矢量中选择一个运动矢量作为输出宏块的运动矢量(称为主运动矢量),典型的准则有:最小距离准则和低频DCT系数用可视化量化矩阵归一化准则。

利用最小距离准则从4个输入运动矢量获得主运动矢量 mv 如下:

$$mv = \frac{1}{2}\arg\left\{\min_{i\in(1,2,3,4)}\sum_{\substack{j=1\\ j\neq i}}^{4} \| mv_i - mv_j \|_p\right\} \tag{10-8}$$

当 $p=2$ 时,其距离为欧式距离 $\| mv_i - mv_j \|_2 = \sqrt{(mv_{ix}-mv_{jx})^2+(mv_{iy}-mv_{jy})^2}$,相应的最小距离准则为最小欧式距离准则[4];

当 $p=1$ 时,其距离为 $\| mv_i - mv_j \|_1 = | mv_{ix}-mv_{jx} | + | mv_{iy}-mv_{jy} |$;

当 $p=\infty$ 时,其距离为 $\| mv_i - mv_j \|_\infty = \max(| mv_{ix}-mv_{jx} |, | mv_{iy}-mv_{jy} |)$,相应的最小距离准则为最大距离最小准则[5]。

综合考虑转码复杂性、重构图像的质量以及编码比特数等因素,最大距离最小准则比其他两种距离准则具有微弱的优势。

低频DCT系数用可视化量化矩阵归一化准则[6] 根据人类视觉特性,将宏块低频DCT系数用可视化量化矩阵(Visual Quantization Matrix, VQM)归一化,选择使归一化后低频DCT系数 ACT_i 最大的运动矢量做为主矢量。ACT_i 的计算方法如下式:

$$ACT_i = \sum_{k=1}^{4}\sum_{m,n=1,2,3}(| DCT_k(m,n) | \times VQM(m,n)) \tag{10-9}$$

其中 $DCT_k(m,n)$ 为第 i 个宏块第 k 个DCT块在位置 (m,n) 的DCT系数,因为人类视觉系统对高频系数不敏感,所以此处只选用DCT系数的前9个较低频系数。通过比较 ACT_i 的大小我们可以得到主运动矢量:

$$mv = \frac{1}{2} mv_i \mid_{ACT_i\text{最大}} \tag{10-10}$$

(3) 主运动矢量精细化方法

在主运动矢量和其他三个运动矢量简单平均构成的搜索窗内,求最小均方差(MSE)所对应的运动矢量。

(4) 运动矢量的加权和方法

对于任意比例的下采样视频转码,一般采用运动矢量的加权和[7] 方法获得输出运动矢量,它特别适用于输入宏块和输出宏块边界不对齐的情况。

2. 帧率改变下的运动矢量映射

视频转码中,当序列的帧率发生改变时,预测编码的参考帧也会发生改变。此时,新的运动矢量往往需要根据相邻多帧图像间运动关系进行重新估算。

假设原输入视频流中连续三帧图像 P_0、P_1、P_2,依次采用帧间预测编码,在转码输出流中帧率发生改变,其中 P_1 在转码时发生跳帧。MB_i 为 P_2 中任一宏块,运动矢量 $mv_{i,2\to1}$ 为宏块 MB_i 从 P_2 到 P_1 的运动矢量。由于 P_1 发生跳帧,在输出视频流中,P_2 的参考帧变为 P_0。宏块 MB_i 从 P_2 到 P_0 的运动矢量 $mv_{i,2\to0}$ 可以通过 MB_i 在 P_1 中的参考宏块的运动信息得到。假设 MB_i 在 P_1 中的参考块 B_{ref} 被 P_1 中相邻的4个宏块 MB_{j1},$j=1,2,3,4$ 所覆盖(如图10-15所示),每个宏块从 P_1 到 P_0 的运动矢量为 $mv_{j1,1\to0}$,$j=1,2,3,4$。在这4个宏块中,若存在某宏块 MB_{k1},满足 B_{ref} 被 MB_{k1} 覆盖的部分最大,这意味着宏块 MB_{k1} 的运动向量占有更大的权重。所以,新的运动矢量可以表示为:

$$mv_{i,2\to0} = mv_{i,2\to1} + mv_{k1,1\to0} \tag{10-11}$$

若4个宏块中,B_{ref} 被4个宏块均匀覆盖,则参考运动矢量 mv 可以由矢量距离最小原则确定:

$$\mathrm{Sum}_j = \sum_{k=1,k\neq j}^{4} \| mv_{j1,1\to 0} - mv_{k1,1\to 0} \| \tag{10-12}$$

$$mv = mv_{k1,1\to 0}, \text{其中 } k = \arg(\min_j \mathrm{Sum}_j) \tag{10-13}$$

$mv_{i,2\to 0}$ 可以表示为：

$$mv_{i,2\to 0} = mv_{i,2\to 1} + mv \tag{10-14}$$

$mv_{i,2\to 0}$ 确定之后还需要进行适当的修正。若 $mv_{i,2\to 0}$ 超出了视频压缩标准所规定的运动范围，则还需要将 $mv_{i,2\to 0}$ 拉回压缩标准所允许的范围内。为了获得更精确的运动矢量，可以进一步以 $mv_{i,2\to 0}$ 为中心进行小范围的运动搜索。进行运动再估计的范围可以根据容许的计算复杂度决策。实验结果表明，对于未超出范围的 $mv_{i,2\to 0}$ 进行 0.5 像素范围内的运动再估计，对于超出范围的 $mv_{i,2\to 0}$ 进行 2 像素范围内的运动再估计，已经可以获得较为理想精度的运动矢量。

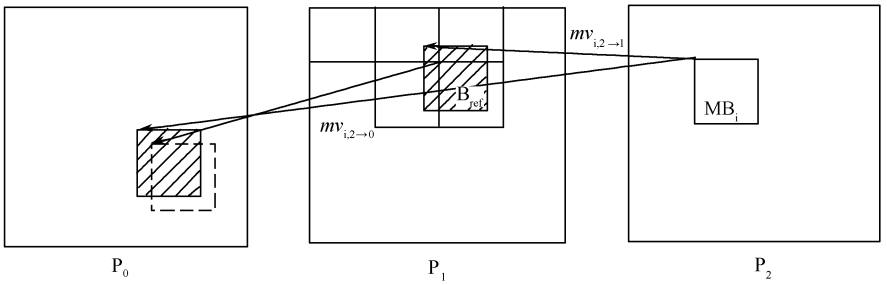


图 10-15　跳帧时运动矢量重用的计算

10.3.3　DCT 域的运动补偿

在 DCT 域做运动补偿可以避免进行费时的 DCT 变换和反 DCT 变换，同时也可以减少缓存开销，从而受到广泛的关注。单就运动补偿而言，在像素域实现运动预测是非常简单的，根据运动向量就可以直接找到预测的像素块。但是对于 DCT 域，因为 DCT 变换是基于块的（通常为 8×8，在 H.264 中为 4×4），所以预测块与参考块之间的 DCT 块通常边界是不对齐的，此时需要将在像素域的处理过程在 DCT 域等价地表示出来。

设 $\hat{X}_1$，$\hat{X}_2$，$\hat{X}_3$ 和 $\hat{X}_4$ 为参考帧中 4 个相邻 8×8 块的 DCT 系数矩阵，其相应的像素值矩阵为 X_1，X_2，X_3 和 X_4，图 10-16 的当前帧中阴影部分块 E 可由参考帧中阴影块 X 预测得到，预测块与 DCT 块边界存在偏移，记水平偏移量为 w，垂直偏移量为 h。则在像素域中，X 可以由 X_1，X_2，X_3 和 X_4 进行相应的矩阵变换得到：

$$X = \sum_{i=1}^{4} C_{i1} X_i C_{i2} \tag{10-15}$$

其中

$$C_{11} = C_{21} = U_h = \begin{bmatrix} 0 & I_h \\ 0 & 0 \end{bmatrix} = \begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}^{8-h} \tag{10-16}$$

$$C_{12} = C_{32} = L_w = \begin{bmatrix} 0 & 0 \\ I_w & 0 \end{bmatrix} = \begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}^{8-w} \tag{10-17}$$

$$C_{31} = C_{41} = L_{8-h}, \quad C_{22} = C_{42} = U_{8-w} \tag{10-18}$$

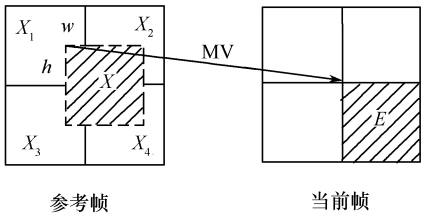


图 10-16　8×8 块运动矢量及参考块的关系

像素块 X 相应的 DCT 系数矩阵 $\hat{\boldsymbol{X}}$ 为：

$$\hat{\boldsymbol{X}} = \boldsymbol{S}\boldsymbol{X}\boldsymbol{S}^{\mathrm{T}} = \sum_{i=1}^{4} SC_{i1}X_iC_{i2}S^{\mathrm{T}} = \sum_{i=1}^{4} SC_{i1}S^{\mathrm{T}}SX_iS^{\mathrm{T}}SC_{i2}S^{\mathrm{T}} = \sum_{i=1}^{4} \hat{C}_{i1}\hat{X}_i\hat{C}_{i2} \tag{10-19}$$

其中

$$\hat{\boldsymbol{C}}_{11} = \hat{\boldsymbol{C}}_{21} = \hat{\boldsymbol{U}}_h = \boldsymbol{S}\begin{bmatrix} 0 & I_h \\ 0 & 0 \end{bmatrix}\boldsymbol{S}^{\mathrm{T}} = (S\begin{bmatrix} 0 & 1 & 0 & \cdots & 0 \\ 0 & 0 & 1 & \cdots & 0 \\ \vdots & \vdots & \vdots & & \vdots \\ 0 & 0 & 0 & \cdots & 1 \\ 0 & 0 & 0 & \cdots & 0 \end{bmatrix}S^{\mathrm{T}})^{8-h} \tag{10-20}$$

$$\hat{\boldsymbol{C}}_{12} = \hat{\boldsymbol{C}}_{32} = \hat{\boldsymbol{L}}_w = \boldsymbol{S}\begin{bmatrix} 0 & 0 \\ I_w & 0 \end{bmatrix}\boldsymbol{S}^{\mathrm{T}} = (S\begin{bmatrix} 0 & 0 & \cdots & 0 & 0 \\ 1 & 0 & \cdots & 0 & 0 \\ 0 & 1 & \cdots & 0 & 0 \\ \vdots & \vdots & & \vdots & \vdots \\ 0 & 0 & 0 & 1 & 0 \end{bmatrix}S^{\mathrm{T}})^{8-w} \tag{10-21}$$

$$\hat{\boldsymbol{C}}_{31} = \hat{\boldsymbol{C}}_{41} = \hat{\boldsymbol{L}}_{8-h}, \quad \hat{\boldsymbol{C}}_{22} = \hat{\boldsymbol{C}}_{42} = \hat{\boldsymbol{U}}_{8-w} \tag{10-22}$$

由于 $\hat{\boldsymbol{C}}_{i1}$ 和 $\hat{\boldsymbol{C}}_{i2}$ 都是常数矩阵，而 DCT 系数矩阵 $\hat{\boldsymbol{X}}_i$ 经过量化和反量化之后，大部分系数都是 0，因此公式(10-19) 可以根据矩阵的稀疏特性寻找相应的快速算法，以减少运算复杂度。

10.3.4　转码的速率控制

视频转码中的速率控制可以采用视频编码中相应的速率控制算法。然而直接采用视频编码中的速率控制算法将存在几个问题：

第一，在速率控制算法中，往往根据 $R-Q$ 非线性模型 $R = S\left(\frac{X_1}{Q} + \frac{X_2}{Q^2}\right)$ 确定量化步长 Q，模型中 S 为编码复杂度，即运动补偿后残差图像像素的绝对值和。在视频编码中，编码复杂度参数 S 可以在进行运动估计时对相应宏块的 SAD 值累加得到。然而在视频转码中往往不再进行运动估计，为了计算 S，必须重新进行一次残差帧图像像素值的绝对值求和，这将增加速率控制算法的计算复杂度。

第二，在视频转码中，原输入视频流中编码比特数、量化参数等编码信息可以更轻松的得到，而这些信息比模型估计值更精确。如果能有效地利用这些已有编码信息，速率控制将精确高效。

基于上述原因，在视频转码中，相关速率控制的改进主要集中在将已有编码信息应用于速率控制相关模型上。我们可以通过下面介绍的 ρ 域速率控制算法比较在视频编码和视频转码中两者的异同。

DCT 系数量化后所需编码速率 R 与 DCT 系数中零系数所占的百分比 ρ 之间存在近似的线性关系。ρ 域速率控制算法利用 R-ρ 线性模型，由欲编码的目标速率 R 确定相应的 ρ 值，然后通过量化参数 QP 与 ρ 之间的 QP-ρ 模型，确定量化参数 QP，从而使编码比特数控制到目标比特数。

1. 视频编码中的 ρ 域速率控制算法[8]

在视频编码中，ρ 域速率控制算法主要分三大步骤：

(1) 预分析

在对新的一帧图像进行编码前，首先需要对其进行预分析，以确定 R-ρ 线性模型以及 QP-ρ 映射表。

确定 R-ρ 线性模型采用两点法，利用已知的两点 (R_1,ρ_1) 和 (R_2,ρ_2)，可以得到

$$R(\rho)=\left(\frac{R_1-R_2}{\rho_1-\rho_2}\right)\rho+\left(\frac{\rho_1R_2-\rho_2R_1}{\rho_1-\rho_2}\right) \tag{10-23}$$

其中，(R_1,ρ_1) 为令 $QP=1$ 时，量化编码后得到的速率以及零系数的百分比，而 (R_2,ρ_2) 为前帧图像编码速率和量化后零系数的百分比。若当前图像是第一帧编码图像，可设 (R_2,ρ_2) 为 $(0,1)$，式(10-23)简化为

$$R(\rho)=\left(\frac{R_{\text{ate}}\mid_{QP=1}}{\rho(1)-1}\right)(\rho-1) \tag{10-24}$$

为了降低计算复杂度，对 (R_1,ρ_1) 的计算也可以采用模型近似得到。例如在 H.263 压缩标准中，当 $QP=1$ 时，DCT 系数编码速率可以近似估计为：

$$R_{\text{ate}}\mid_{QP=1}\cong\frac{1}{N_P}\left(\sum_{x=-25}^{25}\text{BIT}[x/2]\cdot(D_0(x)+D_1(x))+\sum_{i=1}^{N_M}8\cdot\text{INTRA_dc}_i+L\cdot 22\right) \tag{10-25}$$

式中，N_P 为一帧图像的像素数；N_M 为一帧图像的宏块数；x 表示 DCT 系数的值；$D_0(x)$ 和 $D_1(x)$ 分别表示当前帧中帧间编码宏块和帧内编码宏块的 DCT 系数的分布直方图；L 表示 Level 大于 25 和小于 -25 的系数的个数；BIT[] 表示系数为 x 时编码所用的比特数；INTRA_dc_i 表示第 i 宏块中帧内 dc 系数的个数。

QP-ρ 关系可以通过如下公式得到：

$$\rho(QP)=\frac{1}{N_M\cdot 384}\sum_{|x|<2QP}D_0(x)+\frac{1}{N_M\cdot 384}\sum_{|x|<2.5QP}D_1(x) \tag{10-26}$$

通过 QP-ρ 映射关系以及 ρ 值，可以求得相应的 QP。

(2) 帧级速率控制

在 H.263 中，帧级比特分配可以沿用 4.3.3 节所述的 TMN8 算法获得当前帧图像的目标速率，然后利用第一步中得到的 R-ρ 模型求得 ρ 值，然后根据 QP-ρ 模型求得帧级量化参数 QP_{frame}。

(3) 宏块级速率控制

帧级速率控制的误差一般比较大，为了精确地逼近目标速率，往往会进行宏块级的速率控制。在宏块级速率控制中，主要也是完成宏块级的 R-ρ 线性模型、QP-ρ 模型，以及宏块级的目标比特分配等。

每编码一个宏块后，可以根据图像当前的编码信息，对 R-ρ 模型进行更新。设第 i 个宏块编码的比特数为 B_i，DCT 系数中零系数的个数为 z_i，则第 i 个编码速率 R_{i+2} 和零系数百分比 ρ_{i+2} 为

$$R_{i+2}=\frac{B_i}{16\times 16},\quad \rho_{i+2}=\frac{z_i}{16\times 16+8\times 8\times 2}$$

更新后的 R-ρ 模型为

$$\hat{R}(\hat{\rho})=a+b\cdot\hat{\rho} \tag{10-27}$$

其中

$$b=\frac{\sum_{k=1}^{n}R_k\rho_k-\frac{1}{n}\left(\sum_{k=1}^{n}R_k\right)\left(\sum_{k=1}^{n}\rho_k\right)}{\sum_{k=1}^{n}\rho_k^2-\frac{1}{n}\left(\sum_{k=1}^{n}\rho_k\right)},\quad a=\frac{1}{n}\sum_{k=1}^{n}R_k-b\cdot\frac{1}{n}\sum_{k=1}^{n}\rho_k \tag{10-28}$$

$n = i + 2$，(R_1, ρ_1) 和 (R_2, ρ_2) 为预分析所得的两个点。

在编完第 i 个宏块后还需要对 $QP - \rho$ 模型进行更新：

$$\rho_{i+1,N_M}(QP) = \frac{1}{384 \times (N_M - i)}\left[\sum_{|x|<2QP} D^0_{i+1,N_M}(x) + \sum_{|x|<2.5QP} D^1_{i+1,N_M}(x)\right] \tag{10-29}$$

其中 $D^0_{i,j}(x)$ 表示从第 i 个宏块到第 j 个宏块中帧内预测块 DCT 系数的分布直方图，而 $D^1_{i,j}(x)$ 则为第 i 个宏块到第 j 个宏块中帧间预测块 DCT 系数的分布直方图。

当 R-ρ 模型和 QP-ρ 模型完成更新后，可以根据新的模型计算更为精确 $\hat{\rho}$ 和量化参数 QP。为了保证平滑的图像质量，一般还要求量化参数不小于 $QP_{\text{frame}} - 2$。

2. 视频转码中的 ρ 域速率控制算法

在视频编码中，建立 R-ρ 线性模型的两个点 (R_1, ρ_1) 和 (R_2, ρ_2) 一般为编码前一帧得到的 $(R_{\text{prev}}, \rho_{\text{prev}})$ 以及量化参数 $QP = 1$ 时得到的 $(R_{QP=1}, \rho_{QP=1})$。在视频转码中，由于输入视频流已经经过一次量化，R-ρ 关系仅在原量化参数以上部分保持线性关系，因此 $(R_{QP=1}, \rho_{QP=1})$ 对模型已经失效。另外，从输入视频流中能很容易得到当前帧原有的编码信息 $(R_{\text{current}}, \rho_{\text{current}})$，这个编码信息比原模型中前帧信息 $(R_{\text{prev}}, \rho_{\text{prev}})$ 能更准确的反映当前帧的情况。所以，在视频转码的预分析模块中，常采用 $(R_{\text{current}}, \rho_{\text{current}})$ 和量化参数 $QP = 32$ 时得到的 $(R_{QP=32}, \rho_{QP=32})$ 建立 R-ρ 线性模型。

10.4 多描述视频编码

10.4.1 多描述编码的概念

所谓**多描述编码 MDC**(Multiple Description Coding) 是指，一个源信号被编码成几个码流，每一个码流称为一个描述。当由一个源生成的各个描述的码流分别经不同的信道传输时，如果传输是不可靠的，接收端只要收到其中任何一个或几个码流，就可以恢复出低质量的信号；若能接收到全部码流，则可以恢复出高质量的原信号来。这看起来与 SVC 有些类似，不过与 SVC 不同的是，MDC 生成的各个码流具有相同的重要性，而在 SVC 中，基本层码流必须保证可靠传输，如果收不到基本层码流，仅从增强层码流不能重建出任何有用的信号。从以上分析看出，MDC 可以用于对异构终端的服务，能力低的终端只对一个码流解码，能力高的终端对所有码流解码；但是 MDC 最重要的应用场合还是视频在具有多径传输能力的不可靠网络上的传输。由于它与SVC的相似性，我们才将它放在SVC之后、而没有放在第 9 章差错控制中讲述。

图 10-17 给出一个由两个描述的 MDC 的编、解码示意图。图中包含一个编码器和 3 个解码器，编码器产生的两个描述码流分别经两个噪声信道传输。这两个信道可以是收、发端之间的两个独立的物理信道，如无线 Ad hoc 网络中的两条路径；也可以是在收、发端之间只有一个物理信道，但通过时间交错或频率复用形成两个逻辑信道。在后一种情况下，例如在因特网上，在同一信道上交替传送 2 个描述的数据包，如果包长足够大，则可以认为两个逻辑信道上丢包的统计特性是近似独立的。无论是物理信道还是逻辑信道，如果一条路径中断，接收端还能够对从另一条路径接收到的描述解码，得到质量降低的图像。

在解码端，图 10-17 中的三个解码器在一段时间内只有一个在工作。在某个时刻，如果解码端只收到描述 1(或描述 2)，则只有解码器 1(或 2) 工作，重建出一个低质量的信号；如果同时收到 2 个描述，则解码器 0 工作，重建出一个高质量的信号来。我们将解码器 1 和解码器 2 称为**边解码器**，解码器 0 称为**中心解码器**；边解码器和中心解码器的重建失真分别称为**边失真**和**中心失真**。当两个描述的码率相同，且边失真相等时，称为**平衡** MDC。

在图 10-17 中，假设每个描述的速率为 $R_1 = R_2$ 比特 / 样值，总速率 $R = R_1 + R_2$；解码端的边

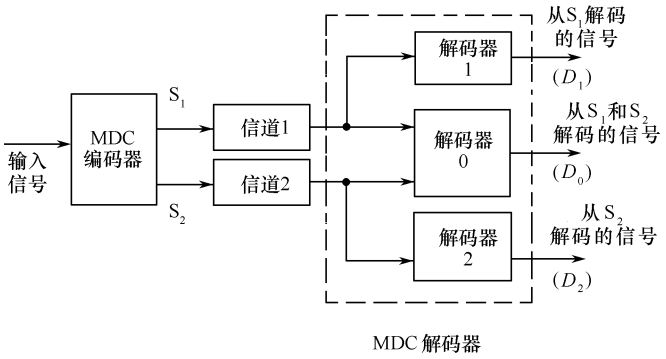


图 10-17　多描述编码与解码

失真为 $D_1 = D_2$，中心失真为 D_0。我们知道，普通的单描述编码器的设计目标是在给定码率 R 下，使失真 D_0 最小；其性能通常由率 - 失真函数 $R(D_0)$ 来度量。对于多描述编码器，我们则需要在给定码率 R 下，同时使中心失真 D_0 和边失真 D_1 最小。这是一个互相矛盾的目标。假设将单描述编码器的输出简单地分成两部分以交替的方式传输来形成 2 个描述，D_0 可以达到最小，而 D_1 则可能大到不能容忍的程度；而在另一种极端情况下，即每一个描述都采用单描述编码器输出的码流，则 D_1 可达到最小，但此时总码率变成 $2R$，中心失真 D_0 要大于速率约束为 $2R$ 的单描述编码码流产生的失真。

为了衡量多描述编码器的性能，我们定义一个**冗余率—失真函数**(Redundancy-Rate Distortion，RRD)。假设一个最好的单描述编码器在速率 R^* 时产生的失真为 D_0，而多描述码编码器在速率 R 时产生的中心失真也为 D_0，则称 $\rho = R - R^*$ 为冗余，它是多描述编码器为降低边失真 D_1 而引入的附加比特，即 RRD 函数 $\rho(D_1; D_0)$ 表示在中心失真为 D_0 时使边失真达到 D_1 所附加的比特数。假设 2 条多描述传输路径发生故障的概率为 $p_1 = p_2 = p$，则多描述编码器的设计目标可以归结为

$$\min_{z,M}[(1-p)^2 D_0 + 2p(1-p)D_1] \qquad (10\text{-}30)$$

$$\text{subject to} \qquad R \leqslant R_{\max}$$

其中 M 代表多描述编码器的参数集，z 代表与之对应的单描述编码器的参数集。D_0 只与 z 有关，而 D_1 与 z 和 M 都有关。将总速率 R 写成 $R = R^*(z) + \rho(M,z)$，式(10-30) 变成

$$\min_{z}\{(1-p)^2 D_0(z) + \lambda R^*(z) + \min_{M}[2p(1-p)D_1(z,M) + \lambda\rho(M,z)]\} \qquad (10\text{-}31)$$

上述优化问题可以看成是一个在最小化平均失真的条件下，如何将总速率 R 在 R^* 和 ρ 之间进行分配的问题。如果 D_0 由应用的需求所限定，则 z 和 R^* 被固定；那么优化问题所需要解决的只是(10-31) 式中大括弧内部的最小化问题。

10.4.2　多描述的生成

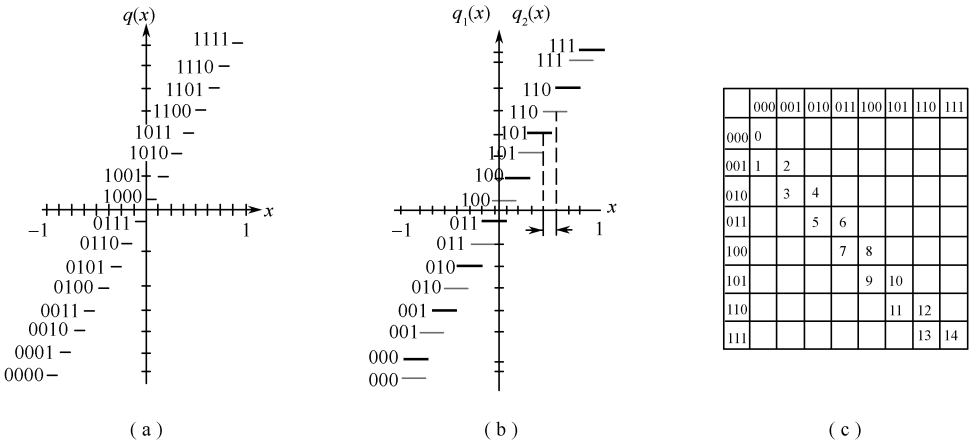
正像 SVC 可以从时间域、空间域或信噪比等不同方面构造不同层次的码流一样，由源信号生成多个描述的方法也有很多，本节将介绍几种典型的方法。要使所产生的多个描述具有相同的重要性，即从每一个描述都可以重建出具有可接受质量的信号，显然每一个描述都需要包含源信号的某些基本信息，因此各个描述之间是相关的。这种相关性使得解码器能够从所接收到的描述去估计丢失的描述，从而提供可接受的重建质量；而在另一方面，描述之间的相关性也增加了总(中心) 码流中的冗余，使得 MDC 的编码效率低于一般的单描述编码。这在(10-30) 式和(10-31) 式的推导过程中我们已经有所认识。

1. 多描述量化

图 10-18(a) 表示一个[−1，1] 区间内的 4b 均匀量化器，在量化电平旁边标注的是对应的 4b 码

字。不管我们采用什么方法，都难以满意地将这个量化器的 4b 输出分成两个描述，因为任何一个描述如果不包含最高比特的话，重建的结果都是很不准确的。图 10-18(b) 给出另一种设计，深色和浅色的横线分别表示两个 3b 量化器的输出电平，两组输出电平是相互交错的。输入信号（图像信号或变换系数）经这两个量化器分别量化可以产生源信号的两个描述，每一个描述给出信号的一种粗量化结果(3b)，而将两个描述结合起来则可增加 1b 的分辨率，得到更精细的量化结果。例如，如果 $q_1(x) = 110$，$q_2(x) = 101$，那么从图看出只收到第 1 个描述时，重建的信号值为 $x = 9/16$；只收到第二个描述，重建 $x = 7/16$；两个描述都收到时，x 必然在 $[7/16, 9/16]$ 内（见图中横向箭头所示区域），因而得到重建值 1/2。在此例中，每个样值需要用 6b 传输（两个信道），而重建的最好质量只与一个 4b 量化器输出相当，其中的冗余是为了保证每一个描述都能提供可接受的重建质量而引入的。

图 10-18(b) 的量化器是如何设计出来的呢？首先假设一个普通的量化器 α，它的输出电平的动态范围被划分成若干个间隔，每个间隔用一个号码标记，我们称这个号码为**索引**(Index)。然后考虑 MD 量化器 α_0，它对每一个输入样值 x 产生一对码字 (i_1, i_2)。在解码端，两个边解码器分别对 i_1 和 i_2 解码，中心解码器对 (i_1, i_2) 解码。图 10-18(c) 给出图(b) 的两个量化器输出码字 i_1 和 i_2 组合所构成的矩阵，矩阵元素中所写的数字 0 ～ 14 就是按 x 值增加排列的 α 的索引值。例如索引值 11 对应的 $(i_1, i_2) = (110, 101)$。在矩阵中为索引指定位置的方法必须是可逆的，即中心解码器可以从 (i_1, i_2) 重建出 α 的输出值。从图(c) 我们看到，64 个矩阵位置只有 15 个被占据，其余的空格代表了冗余信息；而边解码器的重建质量由任何行或列上数字的最大差值（图中为 1）所决定。



	000	001	010	011	100	101	110	111
000	0							
001	1	2						
010		3	4					
011			5	6				
100				7	8			
101					9	10		
110						11	12	
111							13	14

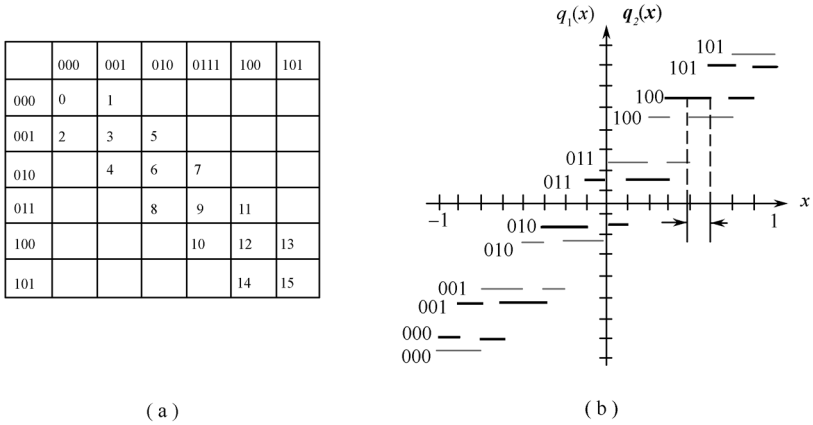
图 10-18　(a) 一般量化器；(b) 一种两描述 MD 量化器；(c) 对应的索引分配

图 10-19(a) 给出另一种索引安排方式，与之对应的 MD 量化器如图(b) 所示。这种索引安排方式比图 10-18(c) 所示的方式冗余度（空格）少，但边解码器的重建质量要差（每行或列上数字的最大差值为 3）。这从图 10-19(b) 所示的 MD 量化器可以更清楚地看到，图中深色和浅色的横线分别代表 2 个量化器的输出电平。相比于图 10-18(b) 的量化器，图 10-19(b) 中的 2 组电平有较少的重叠部分，例如，$q_1(x) = 100$ 意味着 x 处于区间 $[1/4, 3/8] \cup [1/2, 3/4]$，如果再知道 $q_2(x) = 100$ 则意味着 x 处于区间 $[1/2, 5/8]$（见图中横向箭头所示区域）。由于每个量化器只有 6 个输出电平，我们可以称它为 $\log_2 6 = 2.61$b 的量化器。同样达到最好重建质量为 4b 的精度，它比图 10-18(b) 的量化器所用的比特数要少。

有兴趣对 MD 标量量化器设计作进一步了解的读者可参考文献[9]。类似的方法也可以推广到矢量量化器，但由于维数的增多，索引分配的难度和编码的复杂程度大为增加。

2. 空间域或时间域分解

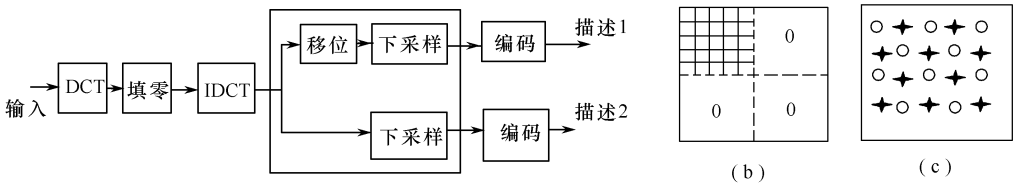
将空域中一部分像素作为一个描述，余下的部分作为另一个描述，然后对每个描述分别编码，



(a) (b)

图 10-19 (a) 另一种索引分配;(b) 对应的 2 描述 MD 量化器

是实现 MDC 的又一种途径。如前所述,为了使边失真足够小,两个描述之间应该存在足够的相关性。直接对一幅图像进行 2∶1 梅花点结构的下采样,可以得到两个尺寸比原图像小的图像作为两个描述。由于图像内容在空间上是连续变化的,这两个描述之间存在一定的相关性。不过为了保证二者之间的相关性足够强,通常我们并不直接对原图像进行下采样,而是首先对它作 2D DCT;然后在 DCT 域内对两个方向填充零,获得一个 4 倍于原图像尺寸的矩阵,并对其进行 2D IDCT;最后对反变换获得的大图像进行下采样得到两个描述,分别进行编码。图 10-20 表示了这个过程,其中图(a)为编码器结构;图(b)为填零后的 DCT 矩阵;图(c)为反变换后的图像矩阵。○和✦分别表示经 2∶1 梅花点结构采样获得的 2 个子图像的像素位置。图(c)中通过交叉取样获得多个子图像的方法常称为**多相分解**(Polyphase decomposition)。从数字信号处理的知识我们知道,在频率域内填零相当于在空间域内进行内插。通过内插,图像更为平滑,增加了两个描述之间的相关性。在接收端,如果收到 2 个描述,则分别将其解码,并将两个解码后的子图像重新组成大图像,然后按上面填零的逆过程,即作 DCT、丢弃填充部分再作 IDCT,恢复出原尺寸的图像来。如果只接收到一个描述,则首先将该描述解码得到一个子图像,然后利用此图像对丢失子图像进行估计,例如用图(c)中的同一列上最邻近的已知像素或上、下、左、右 4 个相邻像素估计丢失像素,得到上采样的大图像;最后按填零的逆过程恢复出原尺寸的图像来。由于这种方法存在编码前填充、解码后又去除填充的过程,因此也称为**预处理—后处理**方法。



(a) (b) (c)

图 10-20 2D 下采样的两描述编码(a) 编码器;(b)DCT 域填零;(c) 空间域分解

另一种在空间域进行分解的方法是,对原图像的每 1 列作 1D DCT,然后如图 10-21(a)所示,在列方向上填充同样长度的零,构成一个长矩阵;对该矩阵的每一列作 1D IDCT,得到 2 倍于原图像高度的一个大图像,最后按行交错地对此图像下采样,得到两个与原图像尺寸相同的子图像;对每个子图像独立地进行图像或视频编码,则产生两个描述。在接收端,如果收到两个描述,则分别对它们解码,并将 2 个解码后的图像重新组合为 2 倍高度的大图像;然后按 1D DCT(列)、丢弃填充部分及 1D IDCT(列)的步骤完成填零的逆过程,恢复出原图像。如果只收到一个描述,则将该描述解码,然后按图 10-21(b)的步骤获得重建图像,其中从一个子图像对另一个子图像进行估值的方法

与 2D 填零时的情况类似。与基于 2D 的 MDC 相比，只在一个方向上填零，使运算量下降；同时由于组合图像比 2D 时的小，提高了中心编码的效率。

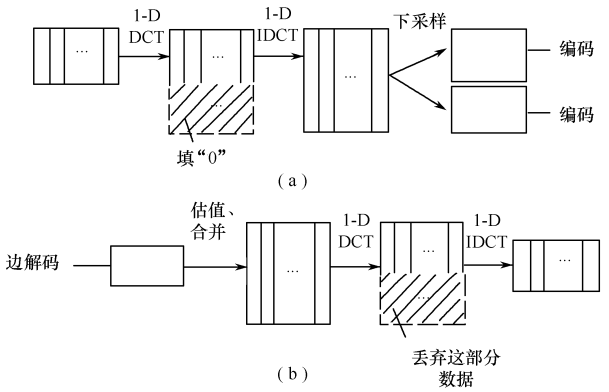


图 10-21　1D 下采样的 2 描述编码，(a) 编码；(b) 边解码

类似的思想也可以应用在对时间域、变换系数域或运动矢量场等的下采样上[10,11]，以得到一个信号的不同描述。此外，上面介绍的填零方法可以看作是一种增加描述之间相关性的预滤波手段，好的预滤波方法有助于在降低边重建失真和提高中心编码效率上取得折中。同时，找到好的分解策略也是值得考虑的问题。

3. 多描述变换编码

如 3.6 节所讲，变换编码通常被用来解除输入样值之间的相关性，变换后的系数是相互独立的。但是在这里，我们希望通过变换编码将输入样值转换成变换系数，以便将变换系数分成几个子集，每个子集独立进行编码，从而构成几个描述的码流。我们已经知道，为了能从接收到的描述估计出传输中丢失的描述，各个描述之间必须存在一定的相关性，相关性的大小决定了重建边失真的大小；而在另一方面，为了在给定的边失真下 MDC 有高的编码效率，每个描述内部的样值之间应该是相互独立(无冗余)的。图 10-22(a) 给出了一个实现上述目标的 2 描述变换编码的一般框架。在该框架中，首先采用 KL 变换消除输入图像(或视频帧)样值之间的相关性，然后通过后一个变换引入达到给定边失真所需要的冗余。后一个变换因此而称为**相关变换**(Correlating Transform)。

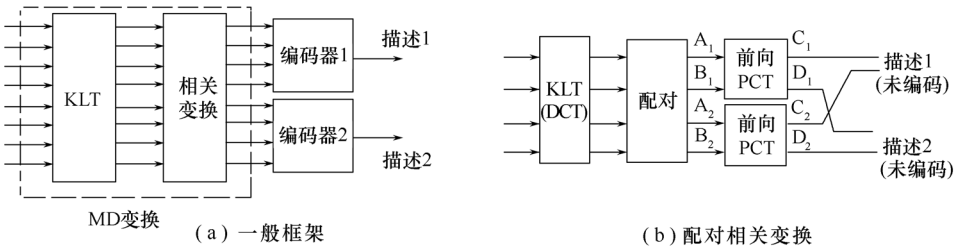


图 10-22　多描述变换编码

图 10-22(b) 是采用一种称为**配对相关变换 PCT**(Pairwise Correlating Transform) 的 MDC 编码器的示意图。在图中，首先对图像块作 $N\times N$(例如 8×8) 的 KL 或 DCT 变换，然后将 N^2 个系数两两配对，得到 $N^2/2$ 个系数对(A,B)，对每个系数对进行如下的相关变换，

$$\begin{bmatrix} C \\ D \end{bmatrix} = \boldsymbol{T}\begin{bmatrix} A \\ B \end{bmatrix} \tag{10-32}$$

其中 $\boldsymbol{T}$ 为 2×2 的变换矩阵。变换后系数对中的两个元素 C 和 D，分别形成两个描述。从上述过程我们可以看出，两个描述之间通过 $\boldsymbol{T}$ 引入了相关性；而由于各个系数对之间是独立的，因此在同一个描述内部样值之间是不相关的。对于图(b) 所示的系统，参考文献[12] 讨论了在给定冗余度下使边

失真最小的最佳变换的具体形式和最佳配对方法等问题，有兴趣的读者可作进一步的阅读。

4. 多描述前问纠错

多描述前问纠错(MD-FEC)是利用9.5.2节讲述的不均衡保护(UEP)来生成多描述码流的一类方法。假设我们将源码流按重要性不同划分成M层(见图10-23)，并将第k层划分成k个等长的部分(或称包)，每层由一个(M,k)RS码来保护，这样每层都产生M个包；然后，取出所有层中的第k个包构成多描述码的第k个描述，共形成M个描述(见图)。由于每层信息包的数量k不同，采用同样长度M的FEC来保护，因此各层的保护力度是不同的。在接收端，只要接收到k个描述(任何k个)，就能利用RS解码恢复出前k层的信息来。在这种方法中，码流的冗余度(也就是边失真)可以通过改变层的划分来控制。当已知接收到k个$(k=1,2,\cdots,M)$描述的概率时，最佳的分层$(R_1,R_2,\cdots,R_M)$可以通过在总码率(包括FEC)固定的约束条件下，最小化期望的失真来得到[13]。

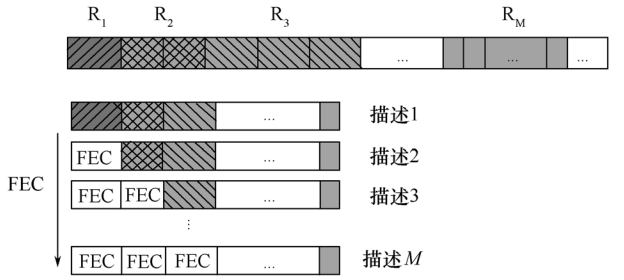


图10-23　多描述前向纠错

从上面的分析可以看出，采用MD-FEC可以从任何形式的可伸缩性编码SVC很容易地生成多描述码流，这使得MD-FEC成为一种常见的生成MD码流的工具。此时，图10-23所示的分层和分流的过程以GOP为单位进行。

10.4.3　视频多描述编码

当将多描述编码应用到视频序列中时，必须注意以下问题。为了提高编码效率，几乎所有的视频编码方案都采用了帧间预测的方法；我们知道在帧间预测编码中，如果收、发两端的参考图像有差异，则会由于失配而产生重建图像的误差，且这种误差还会传播下去，直到出现帧内编码时误差传播才会中止。在MDC中，如图10-17所示，接收端可能接收到2个、也可能只接收到某一个描述，在这两种情况下，接收端所具有的参考图像是不相同的；而发送端对接收端究竟能接收到几个描述并不知情，因此收、发两端参考图像的失配是很可能发生的，这也就决定了失配和误差传播是视频MDC设计中需要考虑的主要问题。

使收、发两端完全不发生失配的一种方法是在发端使用两个预测器，其中每个预测器只针对一个描述进行预测。例如对视频序列在时间上进行2∶1的下采样构成两个描述，每一个描述采用一个一般的单描述编码器编码，则无论接收端收到一个还是两个描述的码流，都不会有失配发生。在发送端只使用一个预测器也有可能不发生失配，此时用来进行预测的参考信息必须同时包含在两个描述之中。一个这样的例子是，用H.263的二层SNR可伸缩性编码产生两个描述，每个描述都包含基本层，而增强层数据则分段交替地放在两个描述中。如果接收端只收到一个描述，则恢复出基本层图像。由于在发送端采用基本层信息进行预测，因此接收端无论收到一个、还是两个描述都不会产生失配。

如果在发送端只采用一个与单描述编码相同的预测器，那么完全没有附加的冗余引入，在接收端收到两个描述时，预测误差最小；但是在接收端只收到一个描述时将发生严重失配。例如，每一个

描述都包含单描述编码器产生的运动矢量和直流(及若干低频)DCT 系数,其余 DCT 系数则分别交替放在两个描述中,那么在接收端接收到两个描述时编码效率与单描述编码一样高,但在只接收到一个描述时边失真很大。

除上述完全不失配和完全失配(无冗余)两种极端情况之外,更多的视频 MD 编码器设计在失配程度和预测效率之间选择适当的折中[14]。此外,在完全失配的方案中还可以在每个描述中增加失配校正信息,以降低边失真。

习 题 十

1. 列举在多媒体信息查询、多媒体会议、VOD,以及数字电视广播等业务中,使用可伸缩性编码的必要性。
2. 画出实现图 10-8(a) 所示的三层可伸缩性编码的编码器的简要方框图。
3. 针对一个未使用可伸缩性编码的一般 MPEG 2 码流,提出两种从已编码流中提取出低质量图像序列子集的方法。查阅 MPEG 2 系统层协议(ISO/IEC 13818—1),并以一个单节目的传送流为例,在计算机上编程实现两种方法的一种。
4. 如果处于高速和低速子网之间的过滤器接收的是打成 RTP 包的 H.264 视频码流,参考第 8 章第 3 题的答案,你认为是否可以对 RTP 包构成的码流进行 QoS 过滤?
5. 根据 10.4.2 节的介绍设计两种产生视频 2 描述码流的方法,(1) 分别说明 2 描述码流的编码过程;(2) 分别分析它是否存在失配;(3) 比较两种方案的失配程度和冗余大小。

参 考 文 献

[1] R Julien, et al., Joint scalable video model JSVM-5. ISO/IEC JTC 1/SC 29/WG 11 N 7796, Jan 2006.

[2] J-R. Ohm, "Advances in Scalable Video Coding," Proc. IEEE, Vol. 93, No. 1, 2005, pp. 42-56.

[3] B. Shen, et al., "Adaptive motion-vector resampling for compressed video downscaling," IEEE Trans. on CSVT, Vol. 9, No. 6, 1999, pp. 929-936.

[4] J. Xin, et al., "An HDTV-to-SDTV spatial transcoder," IEEE Trans. on CSVT, 2002, Vol. 12, No. 1, 2002 pp. 998-1008.

[5] 杜耀刚,蔡安妮,孙景鳌. DCT 域快速下采样运动向量滤波器. 微电子学与计算机, 2005 年第 22 卷第 3 期,pp. 84-88.

[6] A A Yusuf, et al., "An HVS-based motion vector composition algorithm for spatial resolution transcoding,"Inter. Conf. of Information Technology: Coding and Computing (ITCC'04),Vol. 2, 2004, pp. 682-688.

[7] Y. Q. Zhang, et al., "Arbitrary downsizing video transcoding using fast motion vector reestimation[J]," IEEE Signal Processing Letters, Vol. 9, No. 11), 2002, pp. 352 - 355.

[8] Jong-Yun and Hyun Wook Park, "A Rate Control Algorithm for DCT-Based Video Coding Using Simple Rate Estimation and Linear Source Model", IEEE Trans. on CSVT, Vol. 15, No. 9, 2005, pp. 1077-1085

[9] V. A. Vaishampayan, "Design of multiple description scalar quantizers,"IEEE Trans. IT, Vol. 39, pp. 821-834, 1993

[10] C. Kim and S. Lee,"Multiple description coding of motion fields for robust video transmission,"IEEE Trans. CSVT, Vol. 11, pp. 999-1010, 2001

[11] S. Wenger,"Video redundancy coding in H. 263+,"Conf, AV service over packet networks, 1997

[12] Y. Wang, et al.,"Multiple description coding using pairwise correlating transform,"IEEE Trans. IP, Vol. 10, pp. 351-366, 2001

[13] A. E. Mohr, et al. "Unequal loss protection: Graceful degradation of image quality over packet erasure channels through forward error correction,"IEEE JSAC, vol. 18, pp. 819-829, 2000

[14] Y. Wang, et al.,"Multiple description coding for video delivery,"Proc. IEEE, Vol. 93, No. 1, 2005, pp. 57-70